复旦大学管理学院　2026 人工智能领创计划
— 人工智能算法之深度学习

方冠华

统计与数据科学系

教师介绍

- 教育经历：

  复旦大学 数学与应用数学 本科 2011-2015

  哥伦比亚大学 统计学 博士 2015-2020

- 工作经历：

  谷歌 (纽约) 数据科学实习研究员 2019.05 - 2019.08

  百度 (西雅图) 博士后研究员 2020.06 - 2022.06

  阿联酋穆罕默德 · 本 · 扎耶德人工智能大学 访问学者 2022.

  复旦大学 青年副研究员 2022.07 - 至今

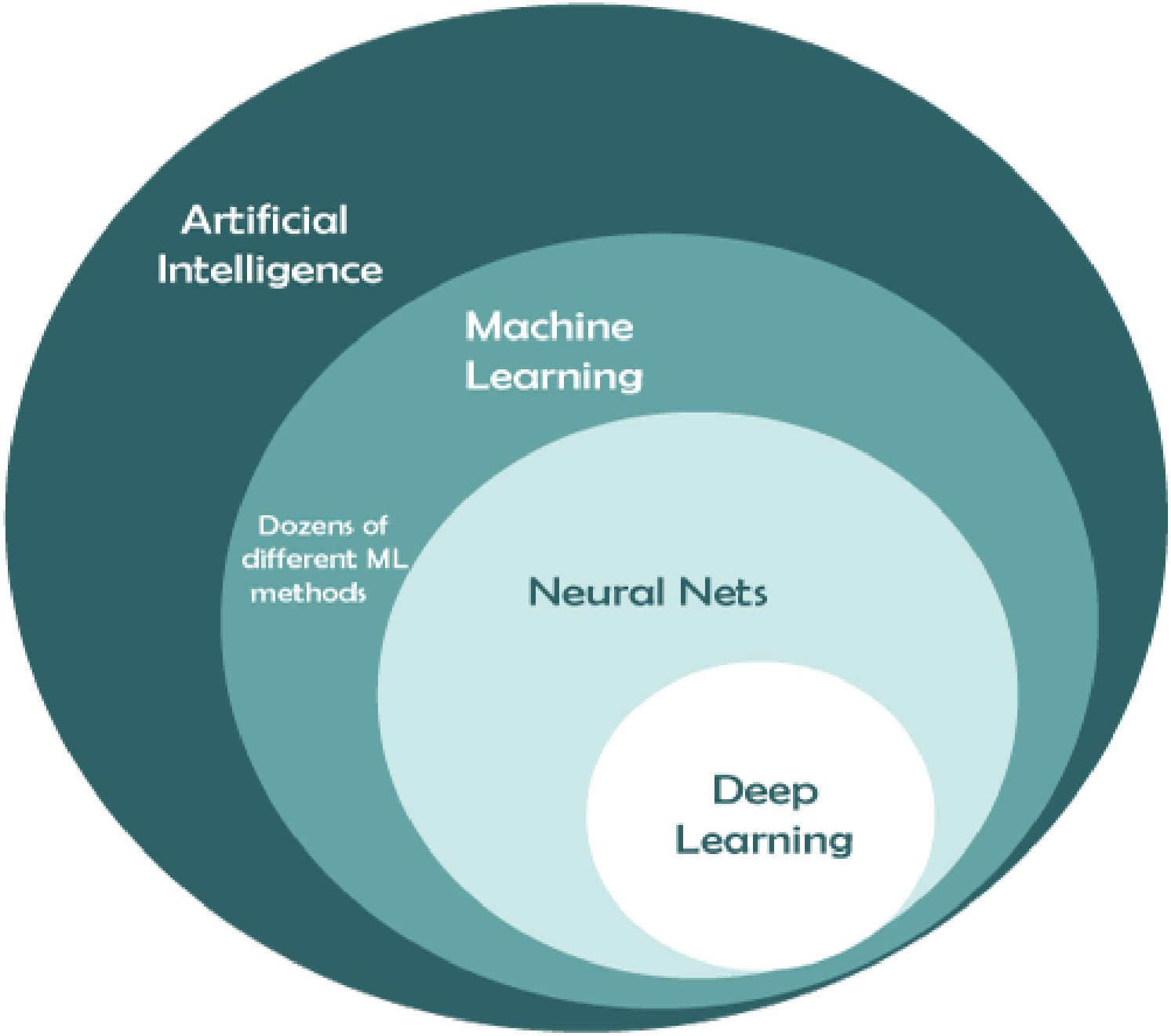
- 研究兴趣

  潜变量模型，机器学习，强化学习
  心理测量，时序过程，大模型

一些基本信息

- 办公室：国顺路 670 号，思源楼 236 室
- 联系方式：fanggh@fudan.edu.cn
- 欢迎随时邮件联系

Appetizers

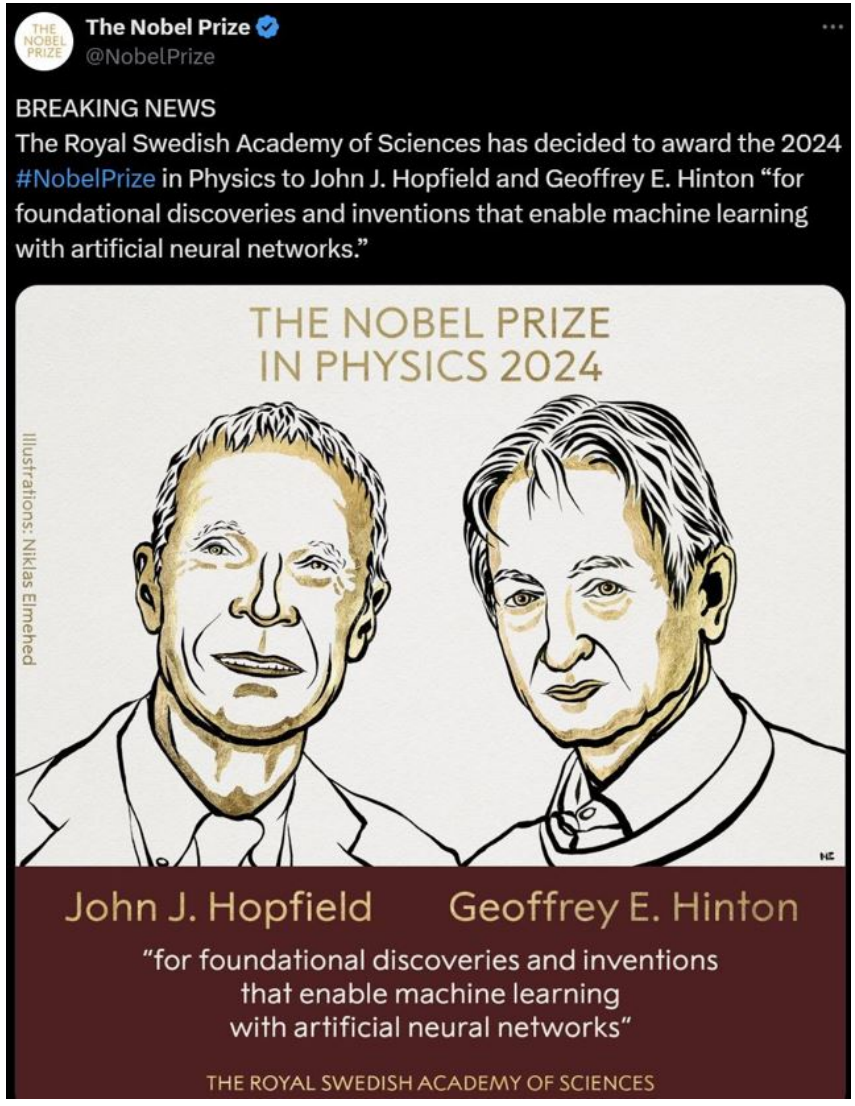

Artificial Intelligence
Machine Learning
Dozens of different ML methods
Neural Nets
Deep Learning

深度学习之父

1. Geoffrey Hinton: 反向传播算法、玻尔兹曼机、AlexNet
2. Yoshua Bengio: 神经网络语言模型、GAN、Theano
3. Yann LeCun: 卷积神经网络、LeNet-5.

诺贝尔物理学奖破天荒颁给“AI 教父”

- Geoffrey Hinton：人类历史上唯一获得图灵奖和诺贝尔物理学奖的科学家

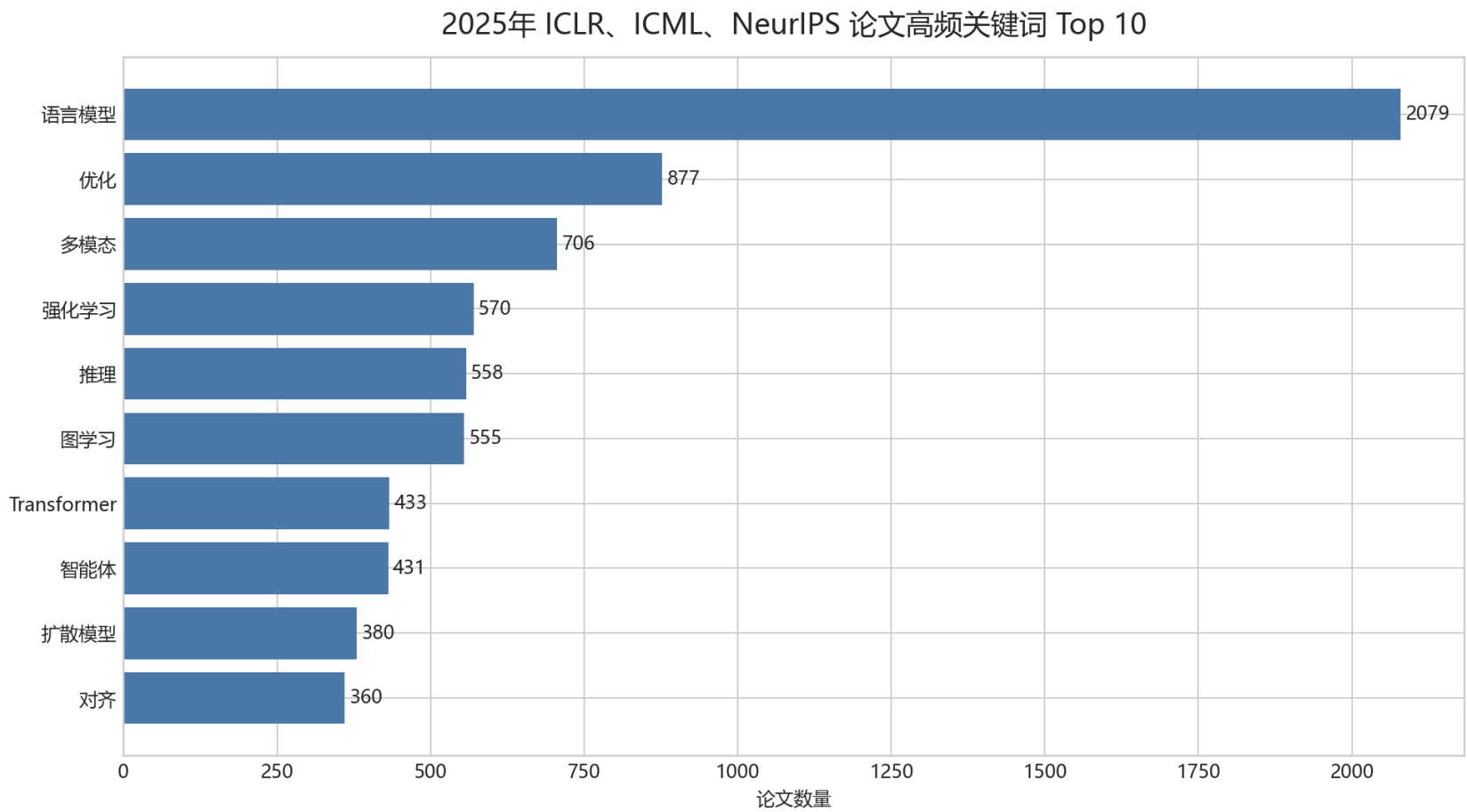


从计算机会议看人工智能发展

- 计算机是一个涵盖面广、选择面宽的领域。因为机器学习、计算机视觉和人工智能领域发展非常迅速，新的工作层出不穷，如果把论文投到期刊上，一两年后刊出时就有点过时了。因此大部分最新的工作都首先发表在顶级会议上，这些顶级会议完全能反映“热门研究方向”、“最新方法”.

五大领域

- **人工智能**: AAAI, IJCAI
- **计算机视觉**: CVPR, ECCV, ICCV
- **自然语言处理**: ACL, EMNLP, NAACL
- **信息检索**: SIGIR, WWW, KDD
- **机器学习**: ICML, NeurIPS, ICLR



数据源：ML Anthology 的 ICLR 2025、ICML 2025、NeurIPS 2025 论文页面
统计方法：基于论文标题的关键词归并；计数表示标题中提到该关键词的论文数

学员们关心的问题

- 没有编程基础?
- 没有统计或 AI 背景?
- 没有阅读最新文献的习惯?

学习的 Roadmap

- 学习基础编码知识、培养数学基本分析能力。
- 学习统计建模、机器学习及深度学习。
  原论文、源代码 –> Vibe Coding 复现
- 专注于一个角色。

机器学习 vs 深度学习

维度	机器学习	深度学习
特征提取	人手动定义（如“苹果是红色”）	机器自动从数据中学习（自己发现“红色 + 圆形 = 苹果”）
数据量	少量数据（几百到几万条）	大量数据（几十万到上亿条）
算法复杂度	简单模型（如决策树、随机森林）	复杂多层网络（如 CNN、RNN）
应用案例	垃圾邮件分类、房价预测	人脸识别、AI 写诗、自动驾驶
类比生活	按菜谱炒菜（人准备食材）	自动种菜 + 做饭（机器自己搞定食材到出锅）

历史发展-1

- (1) 1943 年, 心理学家 McCulloch 和数理逻辑学家 Pitts 提出了神经元的第 1 个数学模型———MP 模型. 它大致模拟了人类神经元的工作原理, 为后来的研究工作提供了依据.
  (2) 1958 年, Rosenblatt 教授提出了感知机模型 (perceptron), Rosenblatt 在 MP 模型的基础之上增加了学习功能, 提出了单层感知器模型, 第一次把神经网络的研究付诸实践. 尽管相比 MP 模型, 该模型能更自动合理地设置权重, 但同样存在较大的局限。
  (3) Marvin Minsky 和 Seymour Papert 于 1969 年证明了感知机模型只能解决线性可分问题, 不能够处理线性不可分问题, 并且否定了多层神经网络训练的可能性, 此后十多年的时间内, 神经网络领域的研究基本处于停滞状态。

历史发展-2

- (4) 20 世纪 80 年代, 计算机飞速发展, 计算能力相较以前也有了质的飞跃。直至 1986 年, Rumelhart 等人在 Nature 上发表文章, 提出了一种按误差逆传播算法训练的多层前馈网络—反向传播网络 (BP 算法), 解决了原来一些单层感知器所不能解决的问题. BP 算法的提出引领了神经网络研究的第二次高潮。
  (5) 由于在 20 世纪 90 年代, 各种浅层机器学习模型相继被提出, 较经典的如支持向量机, 而且当增加神经网络的层数时传统的 BP 网络会遇到局部最优、过拟合及梯度扩散等问题, 这些使得深度模型的研究被搁置.
  (6) 1990 年, LeCun 等提出了现代 CNN 框架的原始版本, 之后又对其进行了改进, 于 1998 年提出了CNN 模型——LeNet-5, 并将其成功应用于手写数字字符的识别中, 1998 年的 LeNet. 最早提出了卷积神经网络，并用于手写数字识别. 只是由于当时缺乏大规模的训练数据, 计算机的计算能力也有限, 所以 LeNet 在解决复杂问题时效果并不好.

历史发展-3

- (7) 2006 年, 机器学习领域泰斗 Hinton 及其团队在 Science 上发表了关于神经网络理念突破性的文章 (Reducing the Dimensionality of Data with Neural Networks), 首次提出了深度学习的概念, 并指明可以通过逐层初始化来解决深度神经网络在训练上的难题。该理论的提出再次激起了神经网络领域研究的浪潮。深度学习的思想再次回到了大众的视野之中, 也正因为如此, 2006 年被称为是深度学习发展的元年。(8) 2012 年, Hinton 教授带领团队参加 ImageNet 图像识别比赛。在比赛中, Hinton 团队所使用的AlexNet 一举夺魁, 其性能达到了碾压第二名 SVM 算法的效果, 自此深度学习的算法思想受到了业界研究者的广泛关注。深度学习的算法也渐渐在许多领域代替了传统的统计学机器学习方法, 成为人工智能中最热门的研究领域.

2022 年之前，公司里为什么需要“古法”码农?

- 如何提高计算效率？
- 我们经常会遇到：

$$\min_{\beta} \|y - X\beta\|^2.$$

- 从理论上来看，求解最优解涉及到计算矩阵的逆矩阵？
- 伴随方法，高斯消去法，LU 分解，Cholesky 分解？

2022 年之前，公司里为什么需要“古法”码农?

- 数据带来的困惑？
- 考虑 0-1 二分类问题：

$$\log\left(\frac{\mathbb{P}(Y=1)}{1-\mathbb{P}(Y=1)}\right)=X\beta.$$

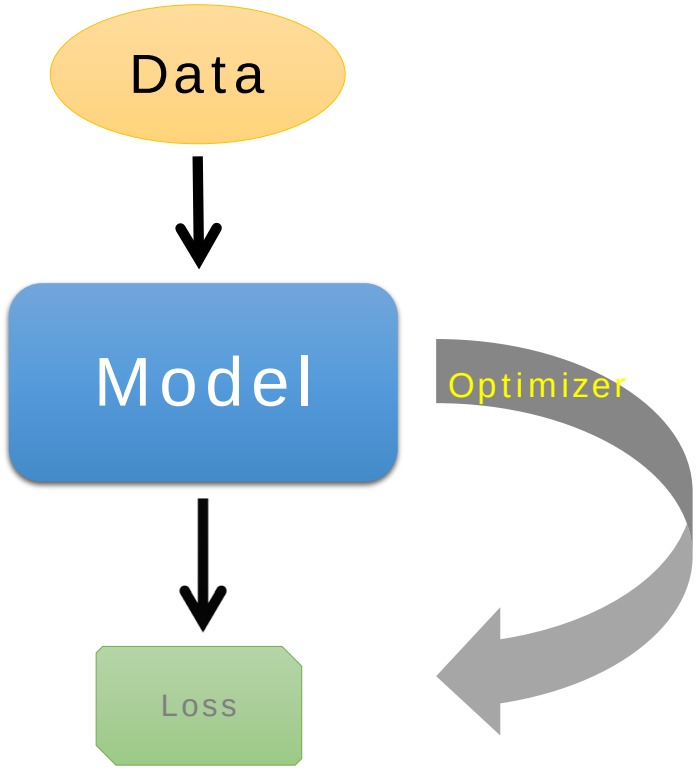
- 若训练样本为：$\{(x_i,y_i)\}$: $(5,1)$, $(4,1)$, $(3,1)$, $(2,1)$, $(1,1)$.
- 或者为：$\{(x_i,y_i)\}$: $(5,1)$, $(4,1)$, $(3,1)$, $(2,0)$, $(1,0)$.
- 则不会有收敛的解。

2022 年之前，公司里为什么需要“古法”码农?

- 计算机中的数值舍入的影响？
- 若直接在 Python 中运行 $\exp(1000)/(\exp(1000)+\exp(1010))$，返回 nan？
- 思考：怎么处理？

Main Courses

四大核心



Data: ImageNet, AIME, 行业垂直领域数据

Model: MLP, CNN, RNN, Transformer, LLM

Optimizer: SGD, ADAM, MUON, GRPO, DPO

Loss: 平方损失，交叉熵损失，正则化损失

模块一：经典神经网络 (MLP) 的底层逻辑

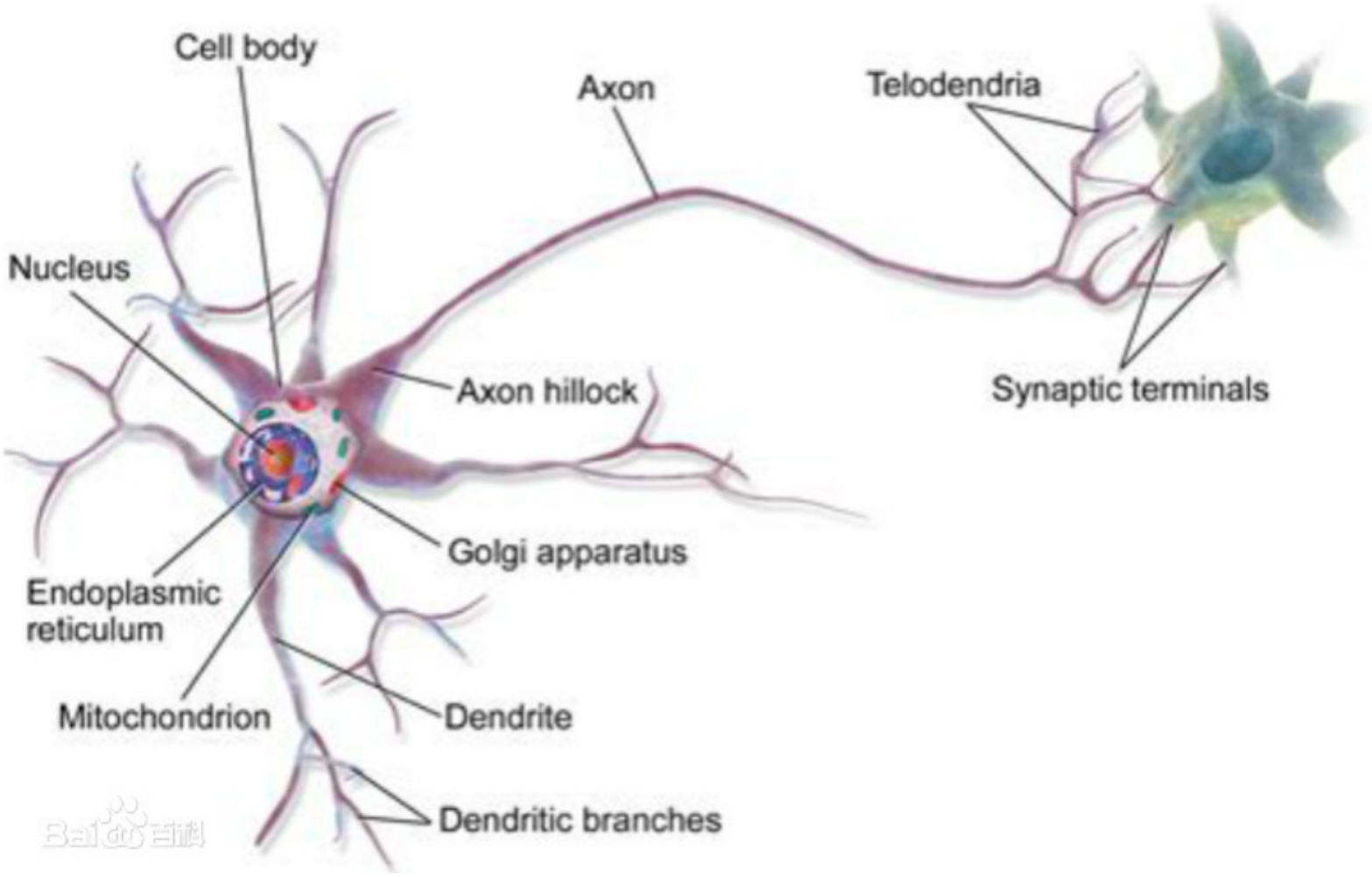
核心技术解码（“数据变现”的引擎）：

- 机制：输入层（各种数据指标）$\rightarrow$ 隐藏层（特征交叉提取）$\rightarrow$ 输出层（概率预测）
- 本质：在海量的、人类无法直观理解的结构化数据（Excel 表）中，寻找隐藏的非线性规律。

神经网络

- 人工神经网络（Artificial Neural Network，ANN）是指一系列受生物学和神经科学启发的数学模型.
- 这些模型主要是通过对人脑的神经元网络进行抽象，构建人工神经元，并按照一定拓扑结构来建立人工神经元之间的连接，来模拟生物神经网络.
- 在人工智能领域，人工神经网络也常常简称为神经网络.

生物神经元


Cell body
Axon
Telodendria
Nucleus
Axon hillock
Synaptic terminals
Golgi apparatus
Endoplasmic reticulum
Mitochondrion
Dendrite
Dendritic branches

神经元

- 人工神经元（Artificial Neuron），简称神经元（Neuron），是构成神经网络的基本单元，其主要是模拟生物神经元的结构和特性，接收一组输入信号并产生输出.
- 生物学家在 20 世纪初就发现了生物神经元的结构.

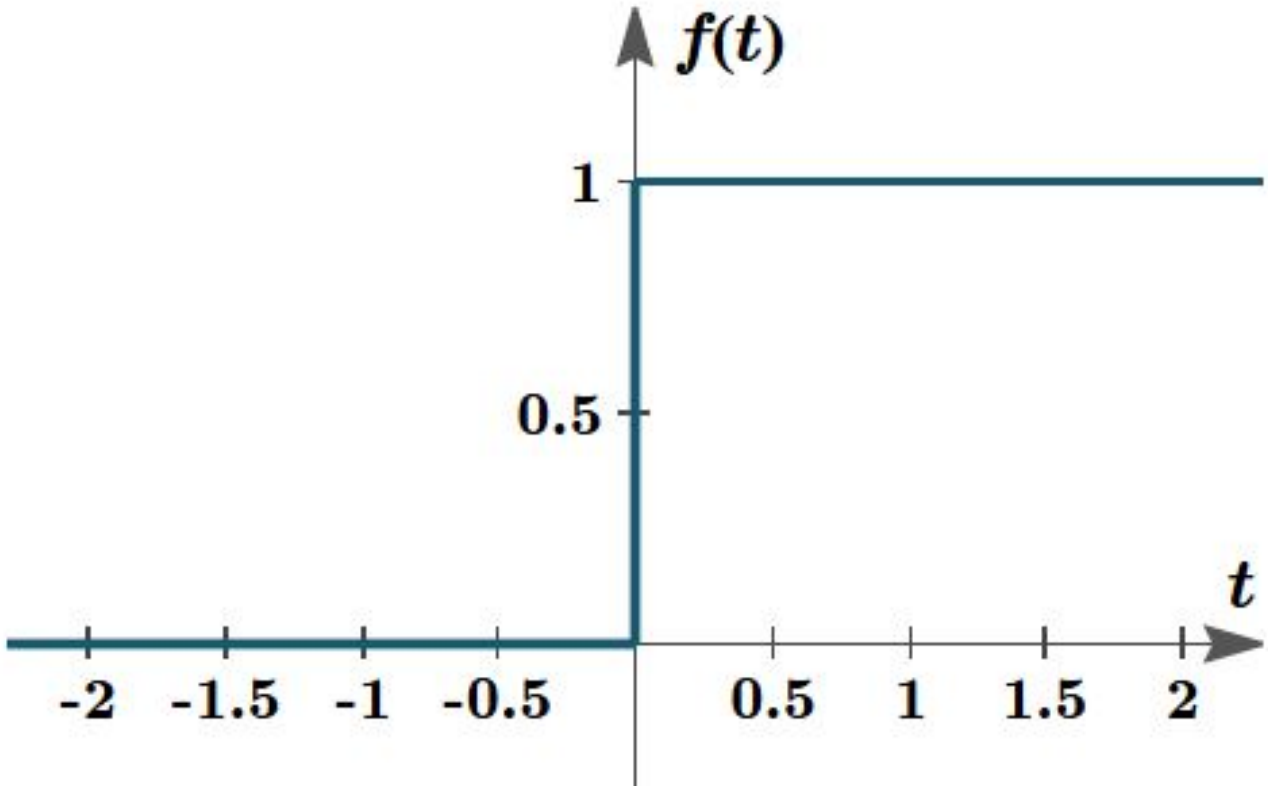
i 一个生物神经元通常具有多个树突和一条轴突. 树突用来接收信息，轴突用来发送信息.

ii 当神经元所获得的输入信号的积累超过某个阈值时，它就处于兴奋状态，产生电脉冲

iii 轴突尾端有许多末梢可以给其他神经元的树突产生连接（突触），并将电脉冲信号传递给其他神经元.

- 1943 年，心理学家 McCulloch 和数学家 Pitts 根据生物神经元的结构，提出了一种非常简单的神经元模型，MP 神经元 [McCulloch et al., 1943].
- 现代神经网络中的神经元和 MP 神经元的结构并无太多变化.
- 不同的是，MP 神经元中的激活函数 f 为 0 或 1 的阶跃函数，而现代神经元中的激活函数通常要求是连续可导的函数.

0-1 的阶梯函数



神经元结构

- 假设一个神经元接受 D 个输入 $x_1, \ldots, x_D$，令向量 $x = (x_1, \ldots, x_D)$ 来表示这组输入，并用净输入 (Net Input) $z \in \mathbb{R}$ 表示一个神经元所获得的输入信号 x 的加权和。
- 具体的，

$$\begin{aligned} z &= \sum_{d=1}^{D} w_d x_d + b \\ &= w^T x + b, \end{aligned} \tag{1}$$

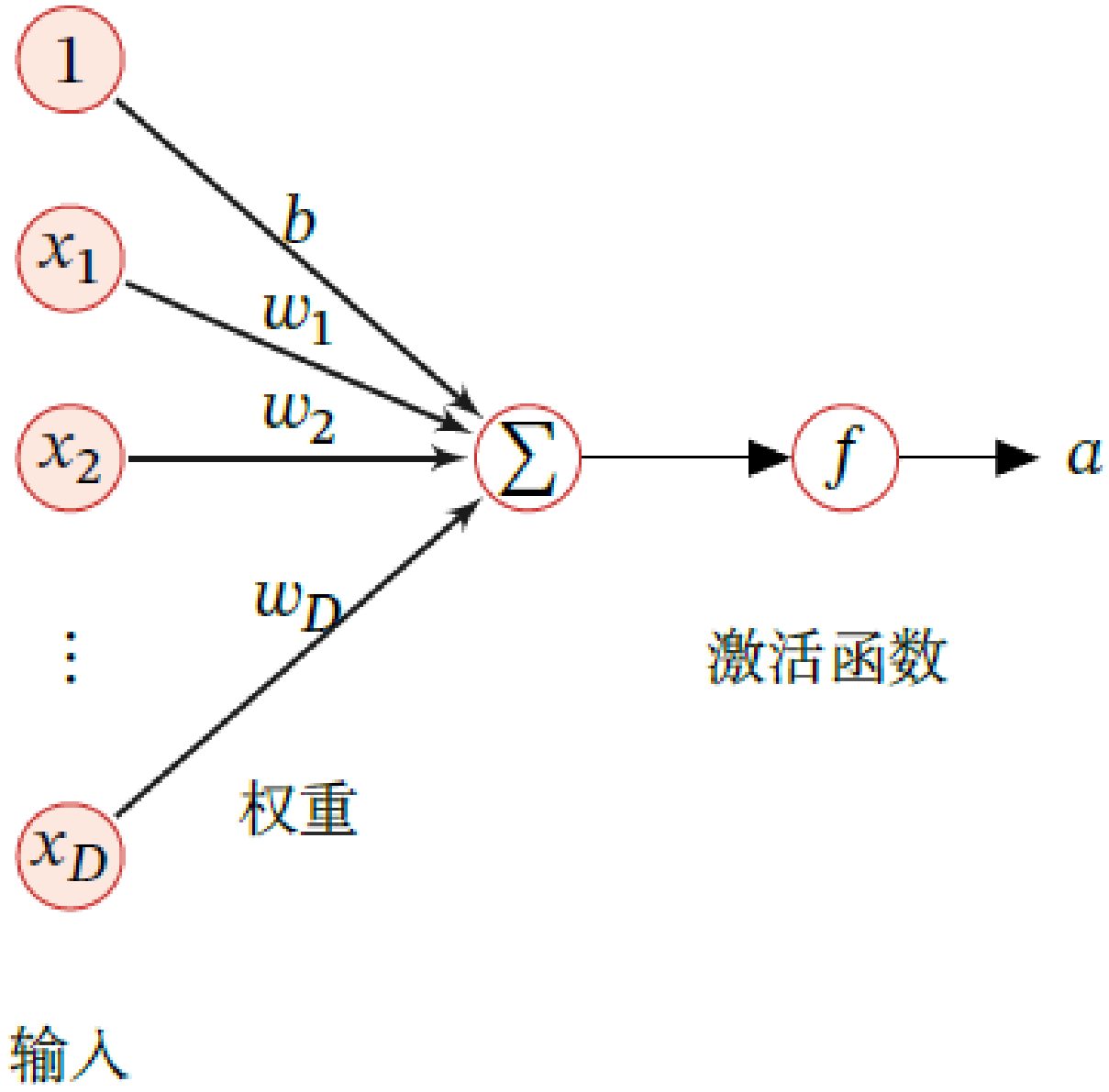
其中 $w = (w_1, \ldots, w_D) \in \mathbb{R}^D$ 是 D 维权重向量，$b \in \mathbb{R}$ 是偏置。

- 净输入 z 在经过一个非线性函数 $f(\cdot)$ 后，得到神经元的活性值 (Activation) a，

$$a = f(z),$$

其中非线性函数 $f(\cdot)$ 称为激活函数 (activation function).

典型的神经元结构图例



激活函数(灵魂)

- 激活函数在神经元中非常重要的. 为了增强网络的表示能力和学习能力，激活函数需要具备以下几点性质:

1. 连续并可导（允许少数点上不可导）的非线性函数. 可导的激活函数可以直接利用数值优化的方法来学习网络参数.
2. 激活函数及其导函数要尽可能的简单，有利于提高网络计算效率.
3. 激活函数的导函数的值域要在一个合适的区间内，不能太大也不能太小，否则会影响训练的效率和稳定性.

Sigmoid 型函数

- Sigmoid 型函数是指一类 S 型曲线函数，为两端饱和函数.
- 常用的 Sigmoid 型函数有 Logistic 函数和 Tanh 函数.
- 数学知识；

  对于函数 $f(x)$, 若 $x \to -\infty$, 其导数 $f'(x) \to 0$, 则称为左饱和。若 $x \to \infty$, 其导数 $f'(x) \to 0$, 则称为右饱和。当同时满足左、右饱和时，就称为两端饱和.

Logistic 函数

- Logistic 函数定义为

$$\sigma(x) = \frac{1}{1 + \exp\{-x\}}.$$

- Logistic 函数可以看成是一个“挤压”函数，把一个实数域的输入“挤压”到 (0, 1).
- 当输入值在 0 附近时，Sigmoid 型函数近似为线性函数.
- 当输入值靠近两端时，对输入进行抑制.
- 输入越小，越接近于 0；输入越大，越接近于 1. 这样的特点也和生物神经元类似.
- 和感知器使用的阶跃激活函数相比，Logistic 函数是连续可导的，其数学性质更好.

Tanh 函数

- Tanh 函数的定义是：

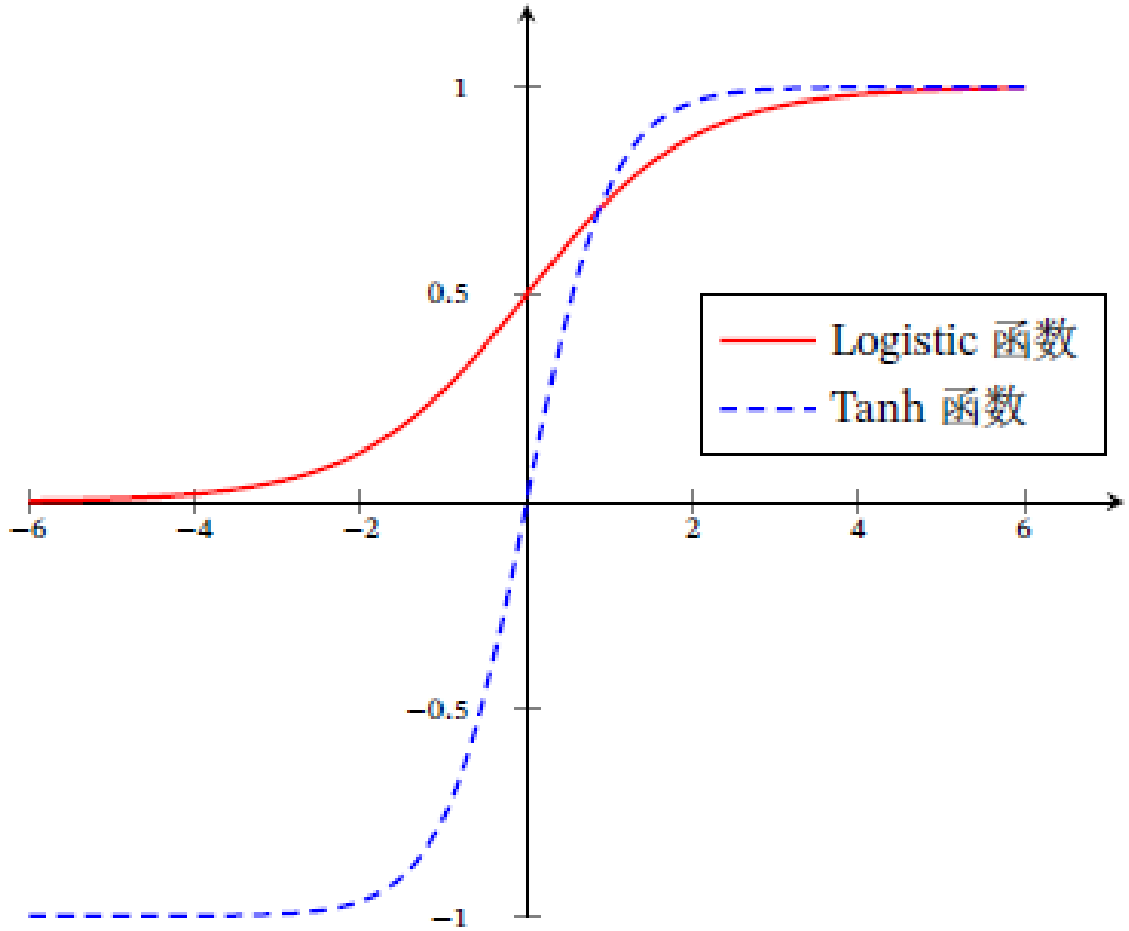
$$\tanh(x) = \frac{\exp\{x\} - \exp\{-x\}}{\exp\{x\} + \exp\{-x\}}.$$

- Tanh 函数可以看作放大并平移的 Logistic 函数，其值域是 $(-1, 1)$.

$$\tanh(x) = 2\sigma(2x) - 1.$$

- 数学性质和 Logistic 完全等价。

Logistic 函数和 Tanh 函数



ReLU 函数

- ReLU（Rectified Linear Unit，**修正线性单元**）[Nair et al., 2010]，**也叫** Rectifier **函数** [Glorot et al., 2011]，**是目前深度神经网络中经常使用的激活函数**.
- ReLU **实际上是一个斜坡**（ramp）**函数，定义为**:

$$\begin{aligned} ReLU(x) &= \begin{cases} x & x \geq 0, \\ 0 & x < 0. \end{cases} \\ &= \max\{0, x\}. \end{aligned} \tag{2}$$

- 优点：

a 神经元只需要进行加、乘和比较的操作，计算上更加高效.

b 具有生物学合理性（Biological Plausibility），比如单侧抑制、宽兴奋边界（即兴奋程度可以非常高）.

c Sigmoid 型激活函数会导致一个非稀疏的神经网络，而 ReLU 却具有很好的稀疏性，大约 50% 的神经元会处于激活状态.

d 一定程度上缓解了神经网络的梯度消失问题

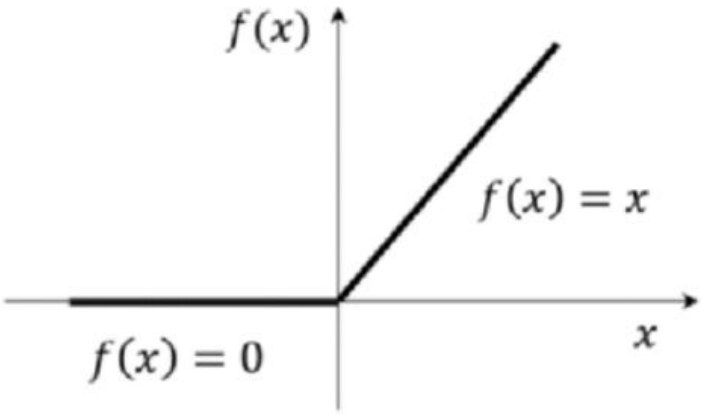
带泄露的 ReLU

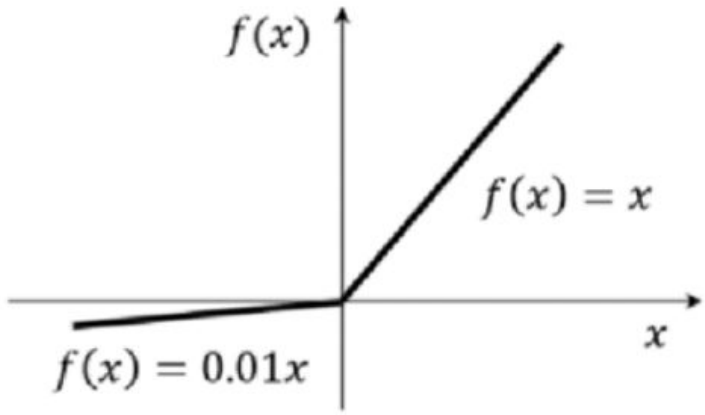
- 带泄露的 ReLU（Leaky ReLU）在输入 $x < 0$ 时，，保持一个很小的梯度 γ.
- 这样当神经元非激活时也能有一个非零的梯度可以更新参数，避免永远不能被激活的情况.
- 其定义如下:

$$\text{LeakyReLU}(x) = \begin{cases} x & \text{if } x > 0, \\ \gamma x & \text{if } x \leq 0, \end{cases} \tag{3}$$

- γ 很小，如 0.01.

ReLU, Leaky ReLU 图像





网络结构

- 一个生物神经细胞的功能比较简单，而人工神经元只是生物神经细胞的理想化和简单实现，功能更加简单.
- 要想模拟人脑的能力，单一的神经元是远远不够的，需要通过很多神经元一起协作来完成复杂的功能.
- 通过一定的连接方式或信息传递方式进行协作的神经元可以看作一个网络，就是神经网络.

基本组成结构

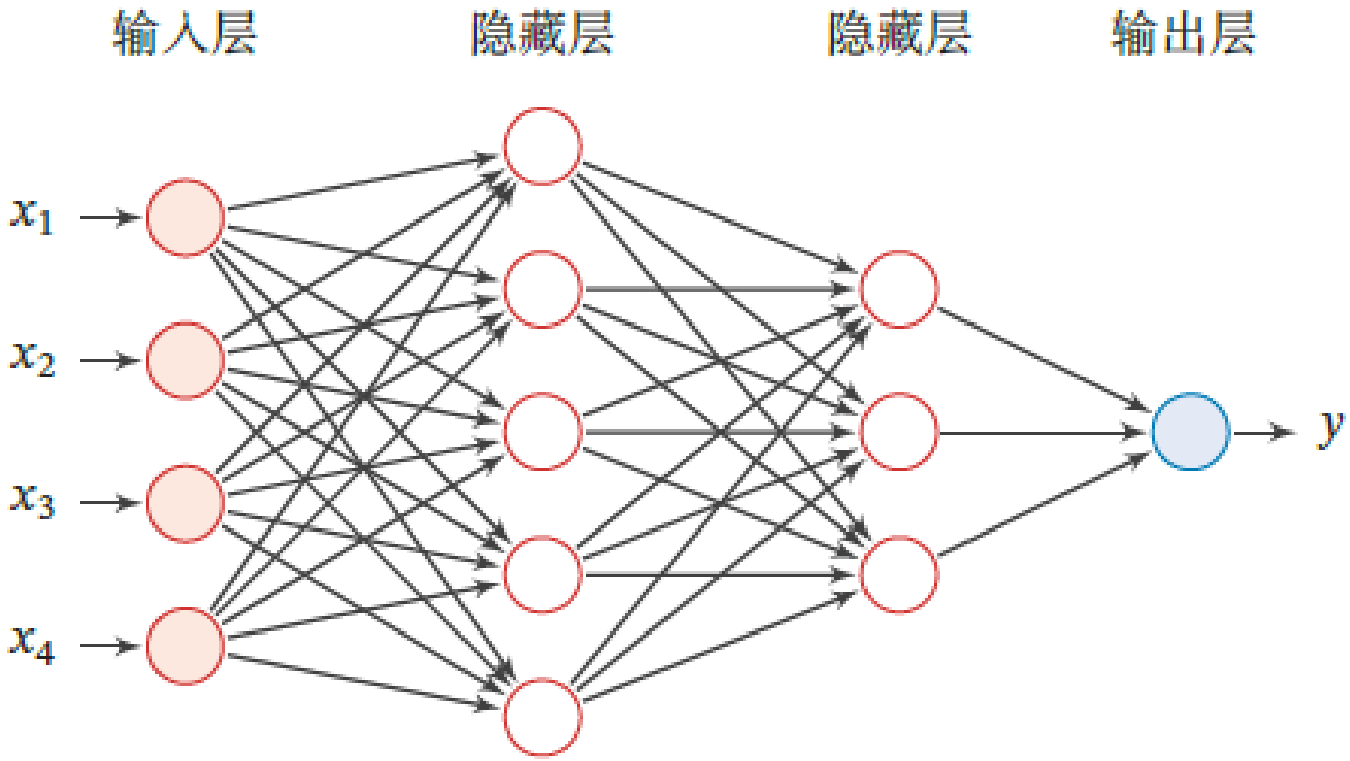
- 一个基本的神经网络由以下几个部分组成：

1. 输入层（Input Layer）：接收输入数据，每个输入节点代表数据集中的一个特征。
2. 隐藏层（Hidden Layers）：一个或多个层次，每层包含若干神经元。这些神经元通过激活函数处理输入信号，并传递给下一层。
3. 输出层（Output Layer）：生成最终的输出信号，其节点的数量通常对应于任务的输出类别或预测值。

前馈神经网络

- 前馈网络中各个神经元按接收信息的先后分为不同的组. 每一组可以看作一个神经层.
- 每一层中的神经元接收前一层神经元的输出，并输出到下一层神经元. 整个网络中的信息是朝一个方向传播，没有反向的信息传播，可以用一个有向无环路图表示.
- 例如：全连接网络，卷积神经网络。
- 前馈网络可以看作一个函数，通过简单非线性函数的多次复合，实现输入空间到输出空间的复杂映射.
- 前馈神经网络要是比作公司，那结构和运作逻辑就像极了一家分工明确的企业，从接收任务到产出结果，层层递进不回头.

多层前馈神经网络图例

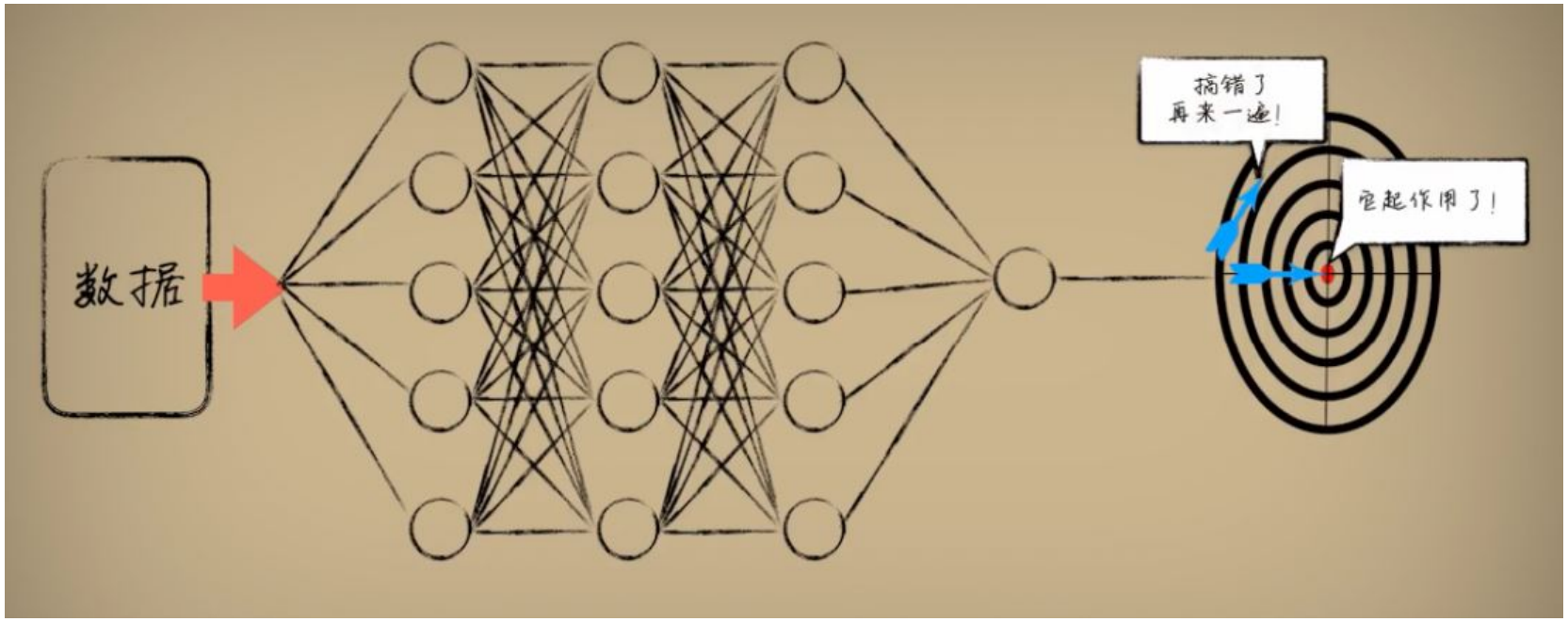


- 在前馈神经网络中，每一层的神经元可以接收前一层神经元的信号，并产生信号输出到下一层.
- 第 0 层称为输入层，最后一层称为输出层，其他中间层称为隐藏层.
- 整个网络中无反馈，信号从输入层向输出层单向传播，可用一个有向无环图表示.

自动梯度计算

- 自动梯度计算神经网络的参数主要通过梯度下降来进行优化.
- 当确定了损失函数以及网络结构后，我们就可以手动用链式法则来计算风险函数对每个参数的梯度，并用代码进行实现.
- 手动求导并转换为计算机程序的过程非常琐碎并容易出错.
- 主流的深度学习框架都包含了自动梯度计算的功能，即我们可以只考虑网络结构并用代码实现，其梯度可以自动进行计算，无须人工干预.

训练(炼丹)



反向传播算法

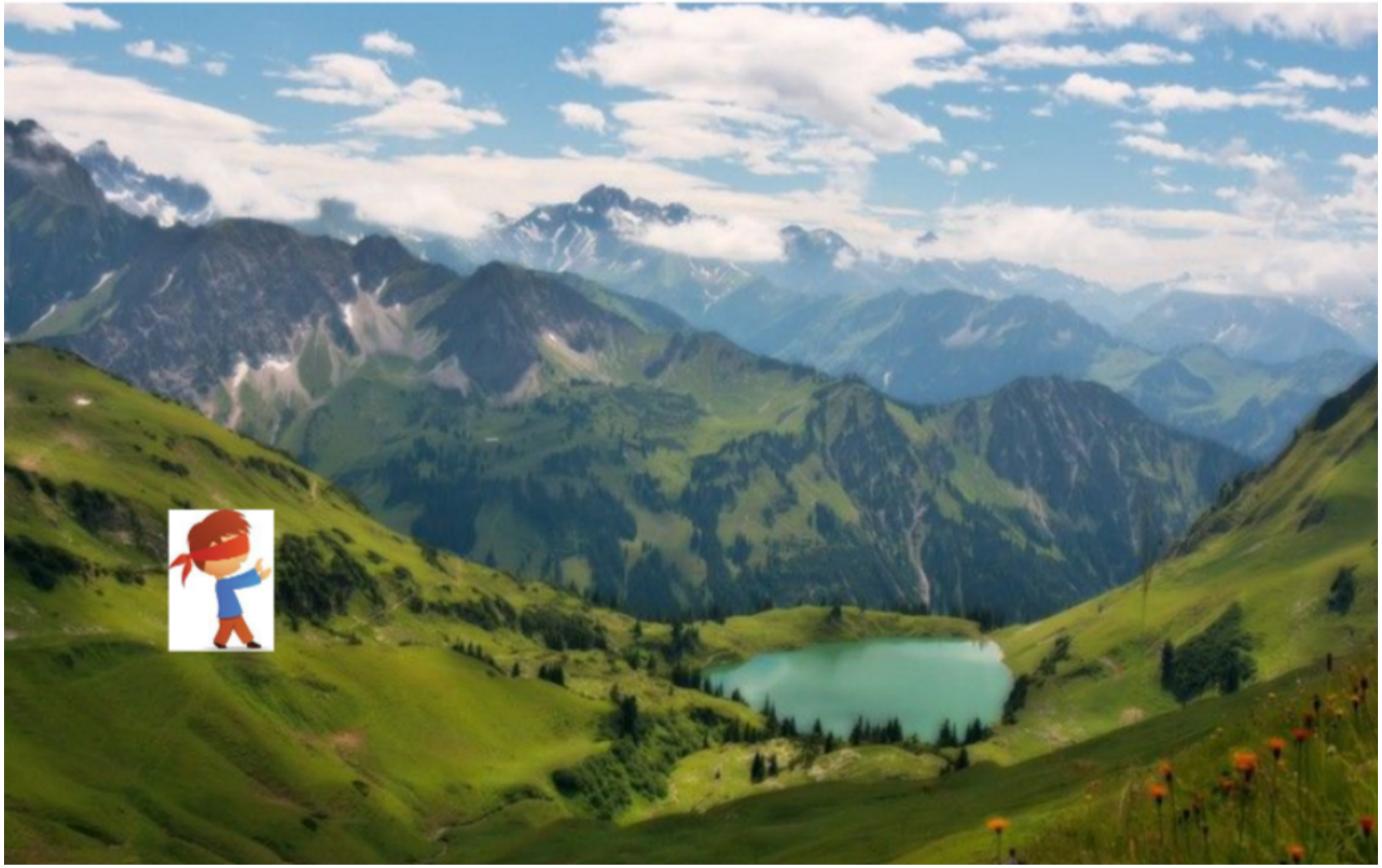
- 假设采用随机梯度下降进行神经网络参数学习.
- 给定一个样本 (x, y)，将其输入到神经网络模型中，得到网络输出为 $\hat{y}$.
- 模型效果越好，$\hat{y}$ 就和 y 越接近。通常的来说，我们会使用损失函数来衡量 $\hat{y}$ 和 y 之间的距离。
- 损失函数可以看成一个打分的“裁判”，看看模型做得有多好或者多差。
- 假设损失函数为 $L(y, \hat{y})$，要进行参数学习就需要计算损失函数关于每个参数的导数.

训练参数步骤

- 使用误差反向传播算法的前馈神经网络训练过程可以分为以下三步:

1. 前向过程: 前馈计算每一层的净输入.
2. 反向过程: 反向传播计算每一层的误差项.
3. 更新过程: 计算每一层参数的偏导数，选取合适的学习率并更新参数.
   新参数 = 旧参数 - 学习率 * 参数梯度.

寻找目标解需要跋山涉水!



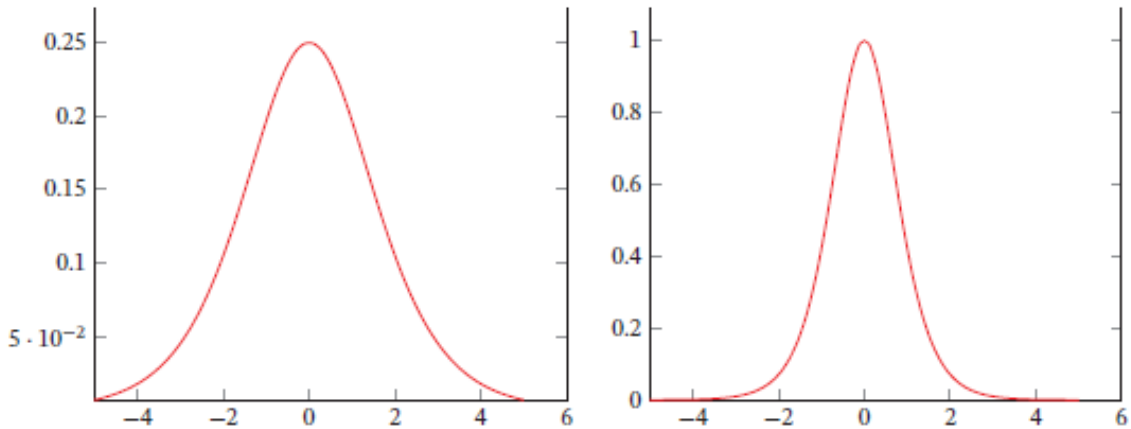
梯度消失问题 (炼丹很难)

- 网络太深!
- Logistic 函数，和 Tanh 函数的导数分别为：

$$\sigma'(x) = \sigma(x)(1 - \sigma(x)) \in [0, 1/4].$$

$$\tanh'(x) = 1 - (\tanh(x))^2 \in [0, 1].$$

- 由于 Sigmoid 型函数的饱和性，饱和区的导数更是接近于 0. 这样，误差经过每一层传递都会不断衰减. 当网络层数很深时，梯度就会不停衰减，甚至消失，使得整个网络很难训练.



(a) Logistic 函数的导数　　(b) Tanh 函数的导数

局部解问题 (炼丹好难)

- 参数太多!
- 低维空间的非凸优化问题是存在局部最优点。
- 高维空间中，非凸优化的难点不在于如何逃离局部最优点，而是如何逃离鞍点 (saddle point).
- 假设网络有 1000 个参数，梯度为 0 的店在某一维是局部最小值的概率维 0.5，那么在独立性的条件下，此点是局部最优点的概率则是 $1/2^{1000}$.

学习率选取问题(炼丹真难)

- 学习率的过大或过小，都会可能使得最终模型效果变差!
- 常见学习率调整算法:

学习率调整	衰减学习率	分段常数衰减，逆时衰减，指数衰减，余弦衰减
	周期性学习率	循环学习率
	自适应学习率	AdaGrad，RMSprop, AdaDelta
梯度估计修正		动量法，Nesterov 加速，AdaDelta
综合方法		Adam

参数的初始化(炼丹太难)

- 神经网络的参数学习是一个非凸优化问题。
- 当使用梯度下降法来进行优化网络参数时，参数初始值的选取十分关键，关系到网络的优化效率和泛化能力。
- 常见初始化方式:
  1. 预训练初始化
  2. 固定值初始化
  3. 随机初始化
  4. 正交初始化

训练模式的展望

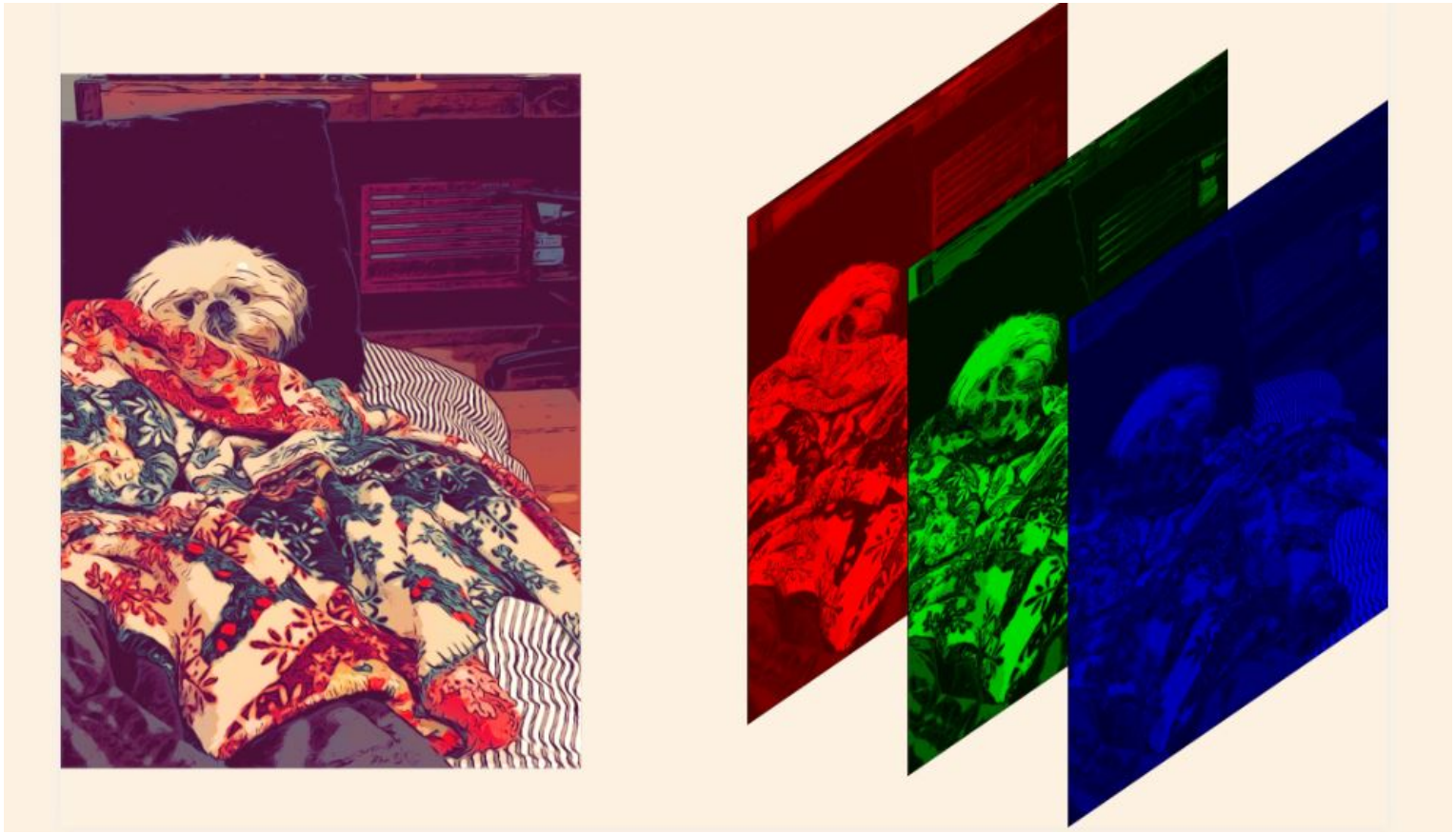
- 过去式: **二阶优化范式** 牛顿法
- 现在式: **一阶优化范式** 梯度下降，随机梯度下降，Adam
- 未来式: **零阶优化范式** 二分法，???

模块二：卷积神经网络 (CNN) 的底层逻辑

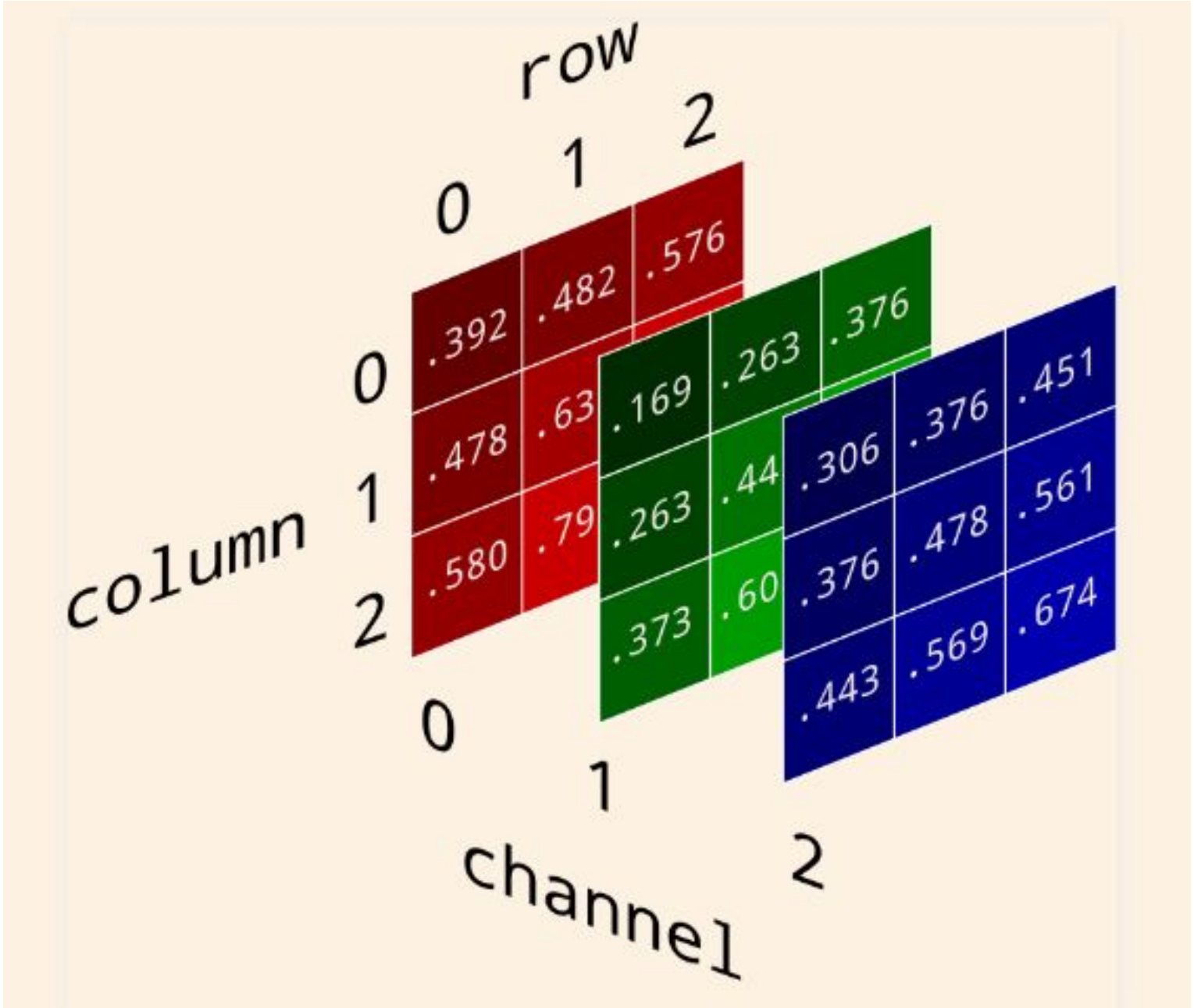
核心技术解码（给机器装上“眼睛”）：

- 机制：通过“卷积核”像手电筒一样扫描图片，从识别点、线、面，到识别复杂物体。
- 本质：让计算机理解空间维度的数据，将现实世界的“视觉信号”转化为“数字资产”。

什么是图片



RGB 表示


row
0
1
2
column
0
1
2
.392
.482
.576
.478
.63
.580
.79
.169
.263
.376
.263
.44
.373
.60
.306
.376
.451
.376
.478
.561
.443
.569
.674
0
1
2
channel

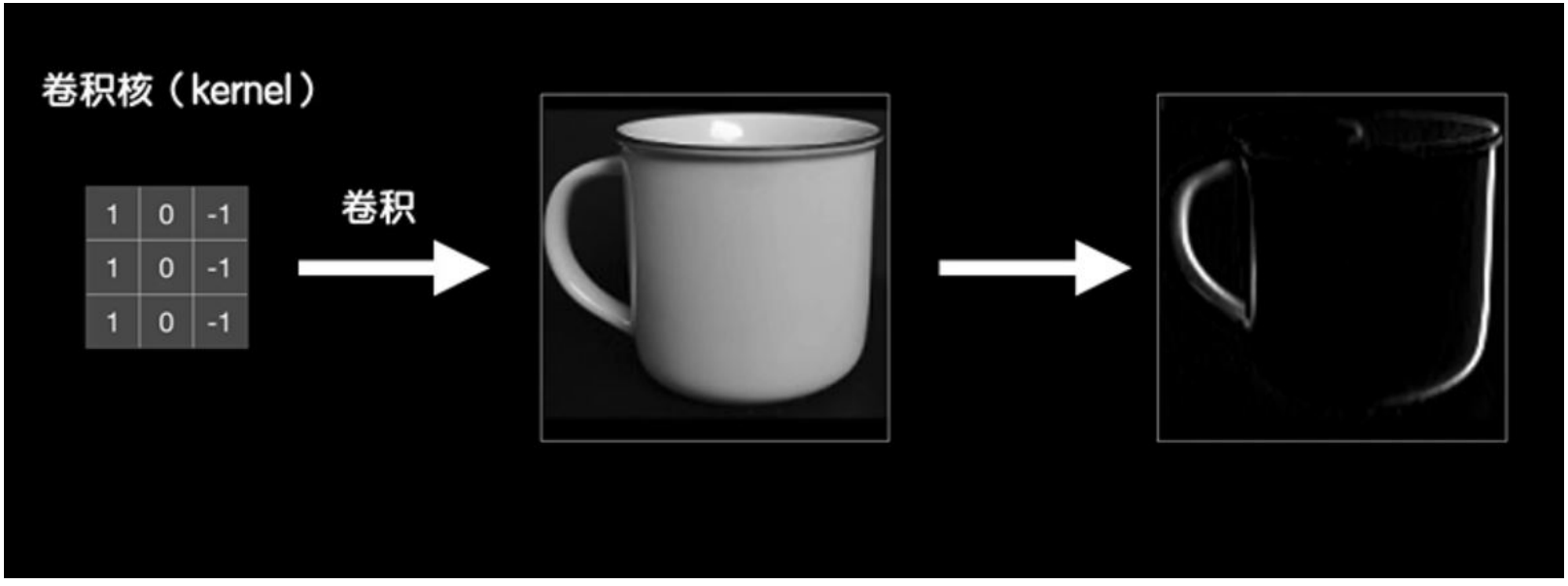
怎么认知图像

- 一个图像本质上是由红 (R)，绿 (G)，蓝 (B) 三个图层叠加起来的矩阵。
- 人在识别一个图片内容的时候，如果看到的是一个个像素拼接起来的数组，那是很难判断图像的内容。
- 实际上人类在识别一个图像的时候，往往会不由自主的提取，比如、颜色、花纹这样的元素。
- 也就是说图像作为一个二维的物体，在二维平面相邻像素之间是存在关联的，我们强行把它降到一维，也就破坏了这些关联，失去了它重要的特征。

特征!

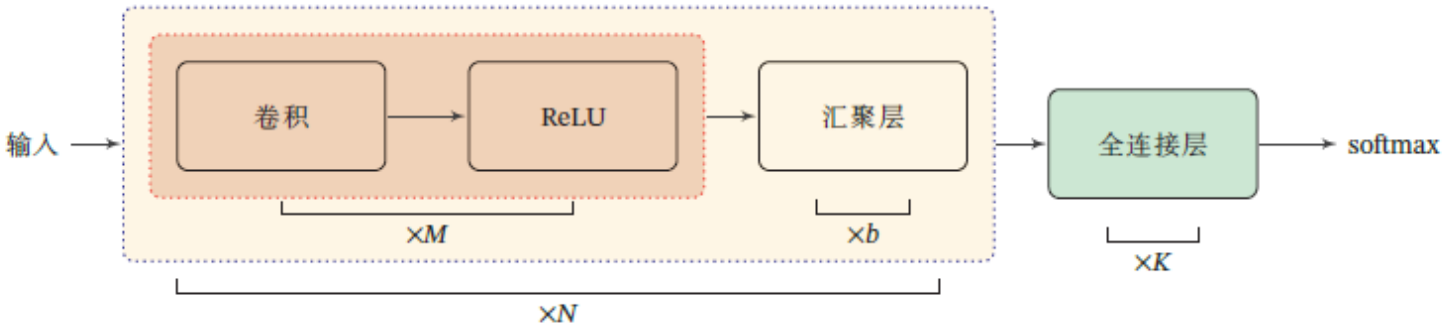
卷积层输入输出

- 输入张量：$X \in \mathbb{R}^{D \times M \times N}$. (例：$D = 1$ 时，输入的是 $M \times N$ 维的灰度图像；$D = 3$ 时，输入的是 $M \times N$ 维的彩色图像)
- 输出张量：$Y \in \mathbb{R}^{P \times M' \times N'}$.
- 卷积核：$W \in \mathbb{R}^{P \times D \times U \times V}$，其中每个切片矩阵 $W^{p,d} \in \mathbb{R}^{U \times V}$，$1 \leq p \leq P$, $1 \leq d \leq D$.


卷积核（kernel）
1 0 -1
1 0 -1
1 0 -1
卷积

卷积网络长啥样呢？

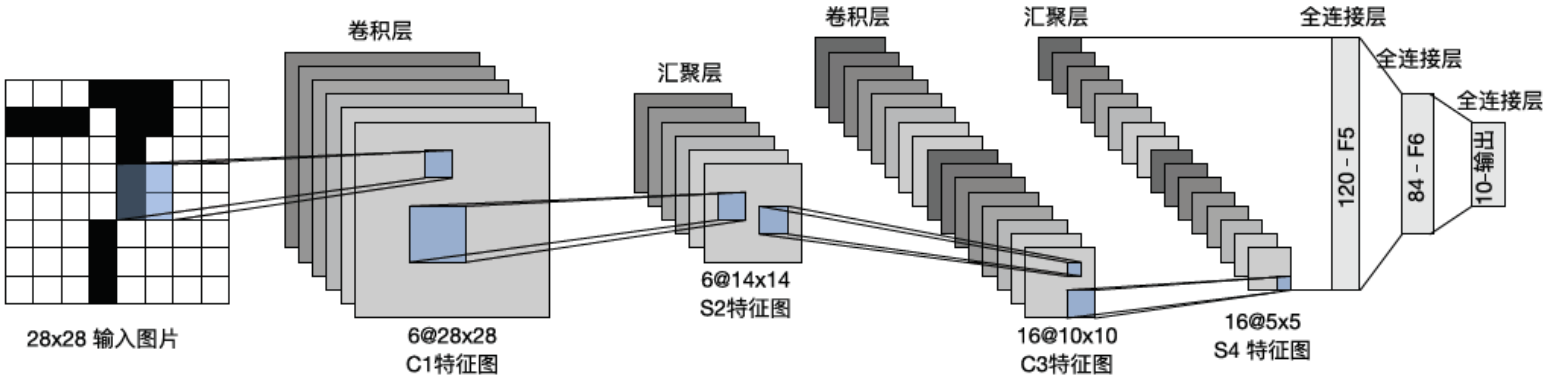
- 一个典型的卷积网络是由卷积层、汇聚层、全连接层交叉堆叠而成. 一个卷积块为连续 M 个卷积层和 b 个汇聚层（M 通常设置为 2 5，b 为 0 或 1）. 一个卷积网络中可以堆叠 N 个连续的卷积块，然后在后面接着 K 个全连接层（N 的取值区间比较大，比如 1 - 100 或者更大；K 一般为 0 - 2）.
- 卷积网络的整体结构趋向于使用更小的卷积核（比如 1×1 和 3×3）以及更深的结构.
- 和建造房子一样，CNN 模型总体的架构是有规律可循的。



LeNet-5

- LeNet-5[LeCun et al., 1998] 虽然提出的时间比较早，但它是一个非常成功的神经网络模型. 基于 LeNet-5 的手写数字识别系统在 20 世纪 90 年代被美国很多银行使用，用来识别支票上面的手写数字.
- LeNet-5 共有 7 层，接受输入图像大小为 $28 \times 28 = 784$，输出对应 10 个类别的得分.
- Demo: https://www.youtube.com/watch?v=FwFduRA_L6Q

LeNet 图示


卷积层
汇聚层
卷积层
汇聚层
全连接层
全连接层
全连接层
120 - F5
84 - F6
10-输出
28x28 输入图片
6@28x28
C1特征图
6@14x14
S2特征图
16@10x10
C3特征图
16@5x5
S4 特征图

- C1 层是卷积层，使用 6 个 5×5 的卷积核，得到 6 组大小为 28×28 = 784 的特征映射. 因此，C1 层的神经元数量为 6 × 784 = 4 704，可训练参数数量为 6 × 25 + 6 = 156.
- S2 层为汇聚层，采样窗口为 2 × 2，使用平均汇聚 (或者最大汇聚)，并通过非线性函数输出. 神经元个数为 6 × 14 × 14 = 1 176，
- C3 层为卷积层. LeNet-5 中用一个连接表来定义输入和输出特征映射之间的依赖关系，共使用 60 个 5 × 5 的卷积核，得到 16 组大小为 10 × 10 的特征映射. (如果不使用连接表，则需要 96 个 5 × 5 的卷积核.) 神经元数量为 16 × 100 = 1 600，可训练参数数量为 (60 × 25) + 16 = 1 516.

- S4 层是一个汇聚层，采样窗口为 2×2，得到 16 个 5×5 大小的特征映射.
- F6 层是一个全连接层，有 120 个神经元，可训练参数数量为 $120 \times (400 + 1) = 48120$.
- F7 层是一个全连接层，有 84 个神经元，可训练参数数量为 $84 \times (120 + 1) = 10\ 164$.
- 输出层：输出层由 10 个径向基函数（Radial Basis Function，RBF）组成.

ILSVRC-ImageNet 竞赛

- ImageNet 是一个超过 15 million 的图像数据集，大约有 22,000 类。
- 是由李飞飞团队从 2007 年开始，耗费大量人力，通过各种方式（网络抓取，人工标注，亚马逊众包平台）收集制作而成，它作为论文在 CVPR-2009 发布。当时人们还很怀疑通过更多数据就能改进算法的看法。
- ILSVRC 是一个比赛，全称是 ImageNet Large-Scale Visual Recognition Challenge，平常说的 ImageNet 比赛指的是这个比赛。
- 一般说的 ImageNet（数据集）实际上指的是 ImageNet 的这个子集，总共有 1000 类，每类大约有 1000 张图像。具体地，有大约 1.2 million 的训练集，5 万验证集，15 万测试集。

https://www.image-net.org/challenges/LSVRC/index.php

计算机视觉三大任务

- 分类、定位、检测!

a 图片分类：给定一张图片，为每张图片打一个标签，说出图片是什么物体，然而因为一张图片中往往有多个物体，因此我们允许你取出概率最大的 5 个，只要前五个概率最大的包含了我们人工标定标签即可。

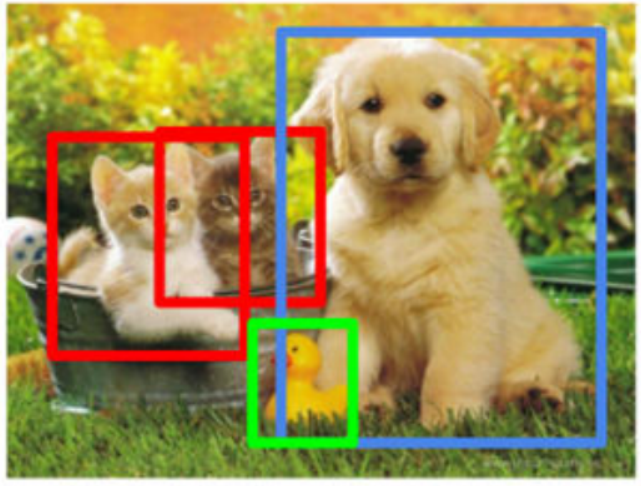
b 定位任务：除了需要预测出图片的类别，还要定位出这个物体的位置，同时规定定位的这个物体框与正确位置差不能超过规定的阈值。

c 检测任务：给定一张图片，把图片中的所有物体全部都找出来 (包括位置、类别)。

Classification

CAT

Object Detection

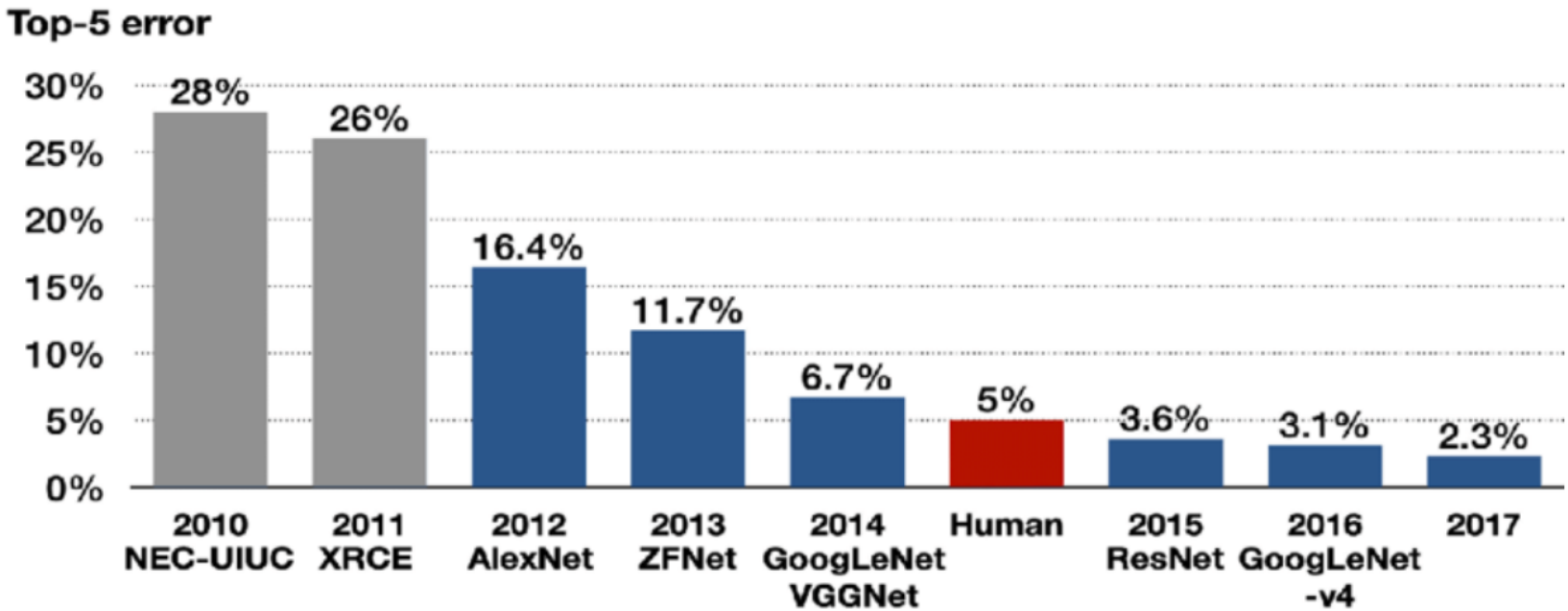


CAT, DOG, DUCK

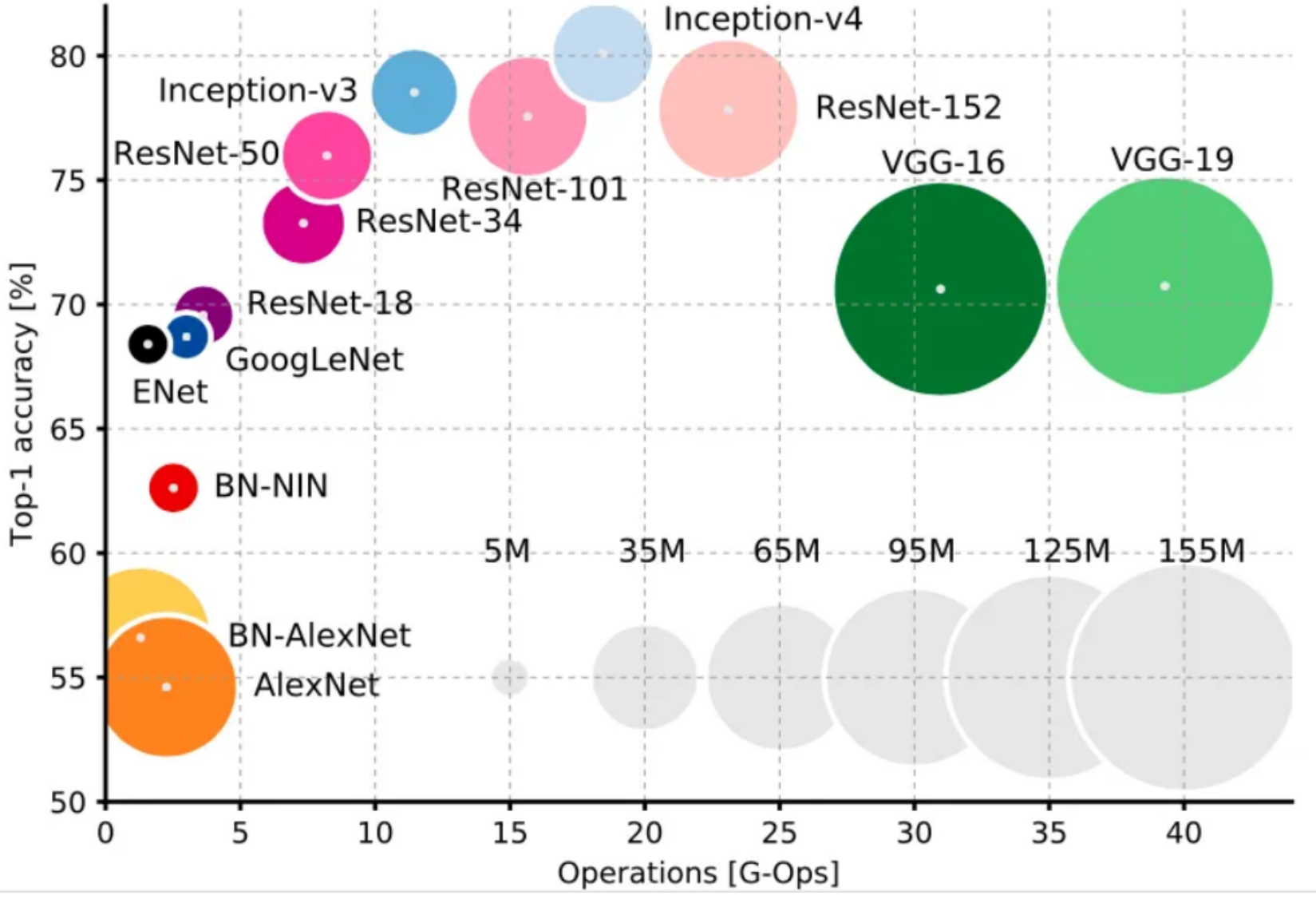
历年冠军

- ILSVRC 从 2010 年开始举办，到 2017 年是最后一届.
- 2012: AlexNet
- 2013: Clarifai, Overfeat
- 2014: GoogLeNet, VGG
- 2015: ResNet
- 2016: Trimps-Soushen
- 2017: SENet

比赛结果比较-1


Top-5 error
30%
25%
20%
15%
10%
5%
0%
28%
26%
16.4%
11.7%
6.7%
5%
3.6%
3.1%
2.3%
2010 NEC-UIUC
2011 XRCE
2012 AlexNet
2013 ZFNet
2014 GoogLeNet VGGNet
Human
2015 ResNet
2016 GoogLeNet -v4
2017

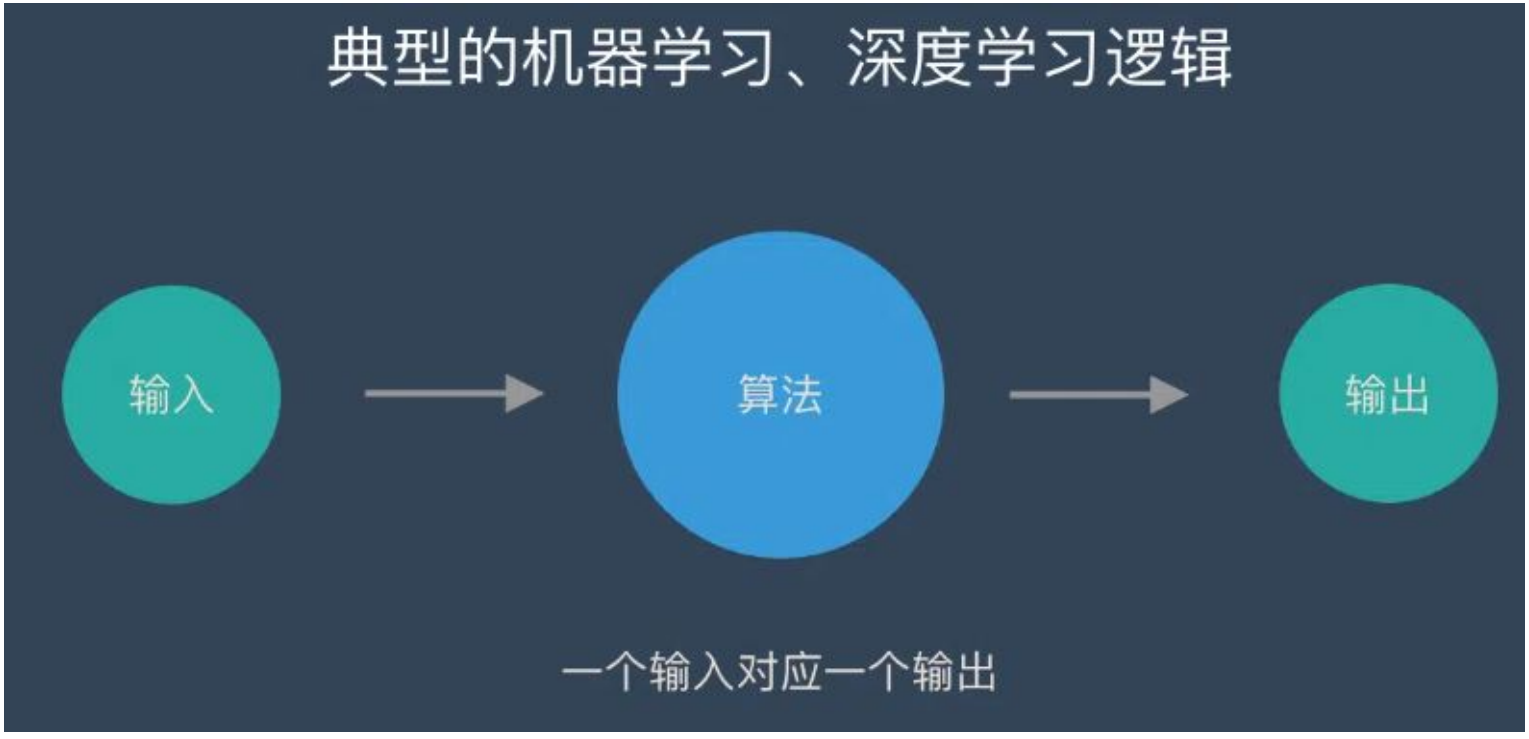
比赛结果比较-2


Inception-v4
Inception-v3
ResNet-152
ResNet-50
VGG-16
VGG-19
ResNet-101
ResNet-34
ResNet-18
GoogLeNet
ENet
BN-NIN
5M
35M
65M
95M
125M
155M
BN-AlexNet
AlexNet
Top-1 accuracy [%]
Operations [G-Ops]
80
75
70
65
60
55
50
0
5
10
15
20
25
30
35
40

模块三：循环神经网络 (RNN) 的底层逻辑

核心技术解码（赋予 AI“记忆”与时间观念）：

- 机制：网络内部包含“循环回路”，前一时刻的状态会影响下一时刻的判断。
- 本质：专门处理“带有时间顺序”的数据（历史影响未来）。


典型的机器学习、深度学习逻辑
输入
算法
输出
一个输入对应一个输出

- 但是在某些场景中，一个输入就不够了！
- 为了填好下面的空，取前面任何一个词都不合适，我们不但需要知道前面所有的词，还需要知道词之间的顺序。

什么是循环神经网络

- 循环神经网络（Recurrent Neural Network，RNN）是一类具有短期记忆能力的神经网络.
- 在循环神经网络中，神经元不但可以接受其他神经元的信息，也可以接受自身的信息.
- 和前馈神经网络相比，循环神经网络更加符合生物神经网络的结构.

与统计的联系

- 在统计里面，自回归模型 (AutoRegressive Model，AR)也可以看成循环神经网络的特例。
- 自回归模型是常用的时间序列模型，用一个变量 y_t 的历史信息来预测自己。

$$y_t = w_0 + \sum_{k=1}^{K} w_k y_{t-k} + \epsilon_t,$$

K 是超参数，$w_0, \ldots, w_K$ 为课学习参数，$\epsilon \sim N(0, \sigma^2)$ 为第 t 时刻的噪声。

循环神经网络的模型结构

- 循环神经网络（Recurrent Neural Network，RNN）通过使用带自反馈的神经元，能够处理任意长度的时序数据.
- 给定一个输入序列 $X_{1:T} = (x_1, \ldots, x_t, \ldots, x_T)$，网络通过下面公式更新隐藏层的活性值 h_t:

$$h_t = f(h_{t-1}, x_t).$$

其中 $h_0 = 0$, $f(\cdot)$ 为一个非线性函数，可以是一个前馈神经网络。

- 隐藏层的活性值 h_t 在文献中也称为隐状态 (hidden state).
- 由于循环神经网络具有短期记忆能力，相当于储存装置，因此其可以逼近任意的非线性系统。

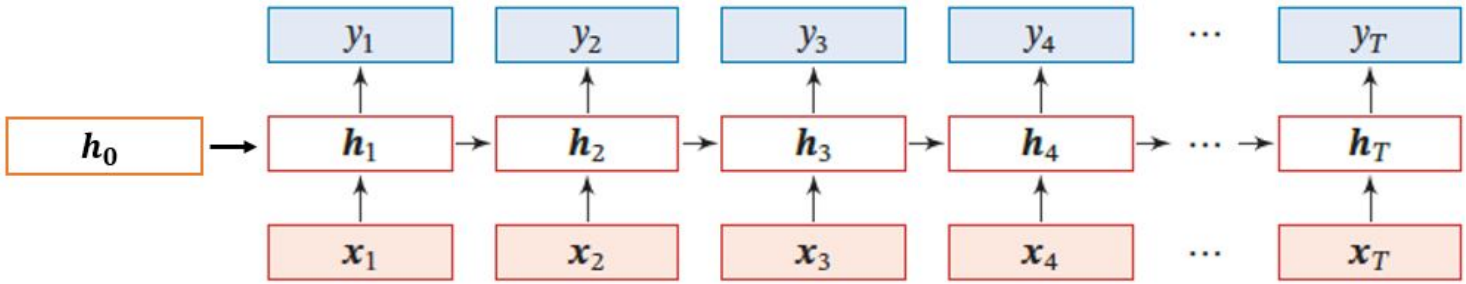
简单循环网络

- 简单循环网络（Simple Recurrent Network，SRN）[Elman, 1990] 是一个非常简单的循环神经网络，只有一个隐藏层的神经网络.
- 令 $x_t \in \mathbb{R}^M$ 表示在时刻 t 网络的输入，$h_t \in \mathbb{R}^D$ 表示隐藏层状态. 则更新公式为：

$$h_t = f(Uh_{t-1} + Wx_t + b),$$

  f 为非线性激活函数，通常为 logistic 或Tanh 或 Relu 函数。
- 通常称 $z_t = Uh_{t-1} + Wx_t + b$ 为隐藏层的净输入。

- 如果我们把每个时刻的状态都看作前馈神经网络的一层，循环神经网络可以看作在时间维度上权值共享的神经网络.



序列到类别模式

- 序列到类别模式主要用于序列数据的分类问题：输入为序列，输出为类别.
- 比如在文本分类中，输入数据为单词的序列，输出为该文本的类别.
- 假设一个样本 $X_{1:T} = (x_1, \ldots, x_T)$ 为一个长度为 T 的序列，输出为一个类别 $y \in \{1, \ldots, C\}$.
- 将样本输入到循环神经网络中，可以的到不同时刻的隐状态，$h_1, \ldots, h_T$.

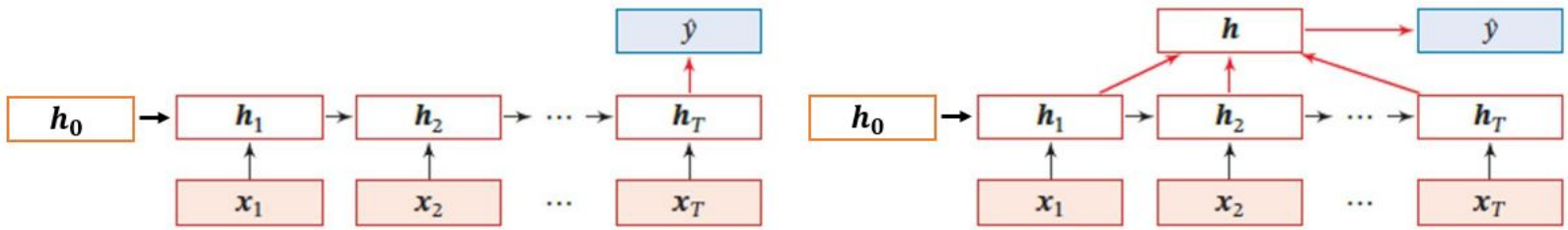
- 可以将 h_T 看作整个序列的最终表示，并作为最终分类器的输入，即

$$\hat{y} = g(h_T),$$

其中 g 是个简单的分类器。

- 也可以将整个序列的平均隐状态看成序列的最终表示，并作为最终分类器的输入，即

$$\hat{y} = g(\frac{1}{T}\sum_t h_t).$$



(a) 正常模式　　(b) 按时间进行平均采样模式


My experience so far has been fantastic!
POSITIVE

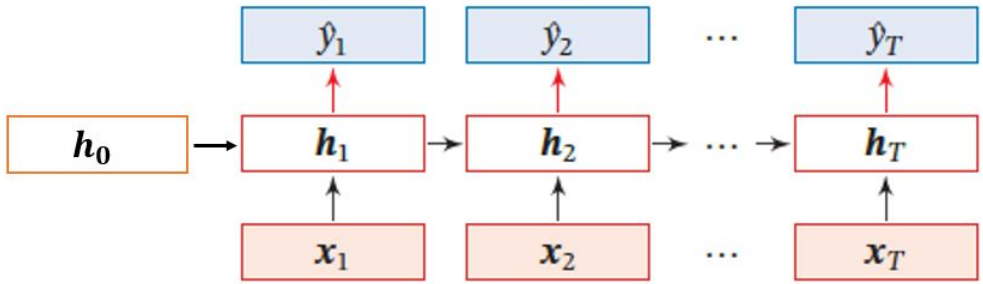

The product is ok I guess
NEUTRAL


Your support team is useless
NEGATIVE

同步的序列到序列模式

- 同步的序列到序列模式主要用于序列标注（Sequence Labeling）任务.（词性标注）
- 即每一时刻都有输入和输出，输入序列和输出序列的长度相同.

$$\hat{y}_t = g(h_t).$$



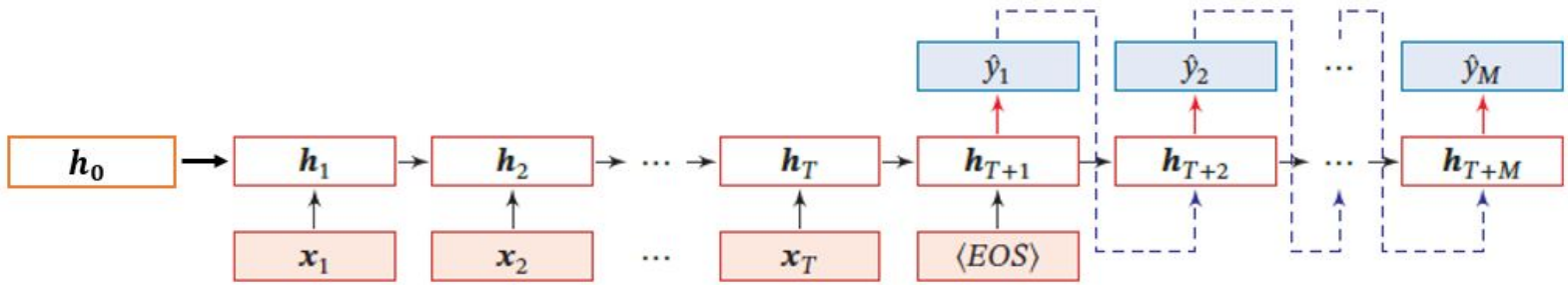

如：输入序列：
新任总裁罗建国宣布了对部门经理邓奇的任免通知
输出序列：
0 0 0 0 B I E 0 0 0 0 0 0 0 0 B E 0 0 0 0 0

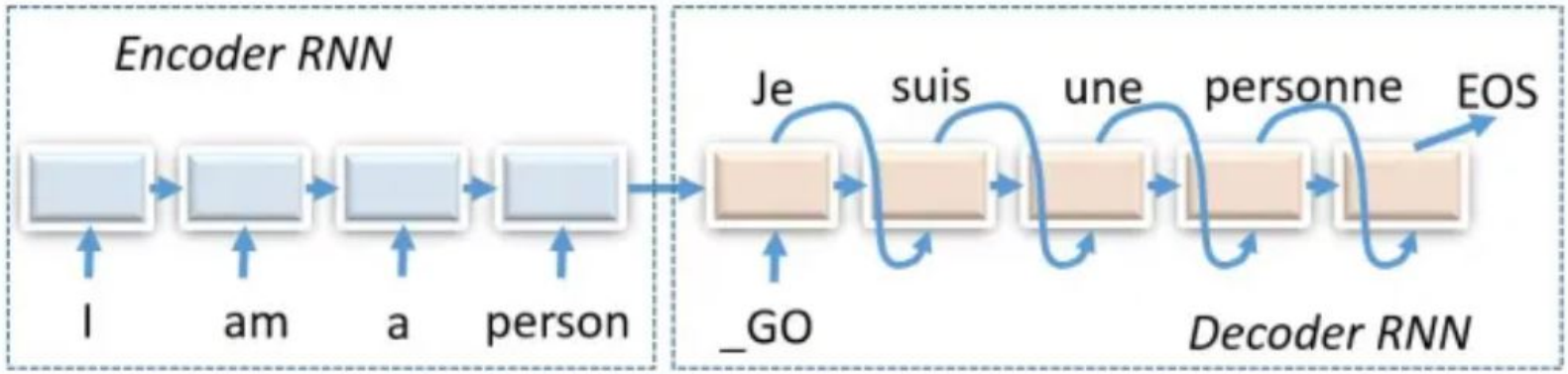
异步的序列到序列模式

- 异步的序列到序列模式也称为编码器-解码器（Encoder-Decoder）模型.
- 即输入序列和输出序列不需要有严格的对应关系，也不需要保持相同的长度.
- 比如在机器翻译中，输入为源语言的单词序列，输出为目标语言的单词序列.

- 输入为长度为 T 的序列，$X_{1:T} = (x_1, \ldots, x_T)$，输出为长度为 M 的序列 $y_{1:M} = (y_1, \ldots, y_M)$.
- 先将样本 x 输入到编码器中，得到编码 h_T. 然后再使用另一个循环神经网络，得到输出序列 $\hat{y}_{1:M}$.

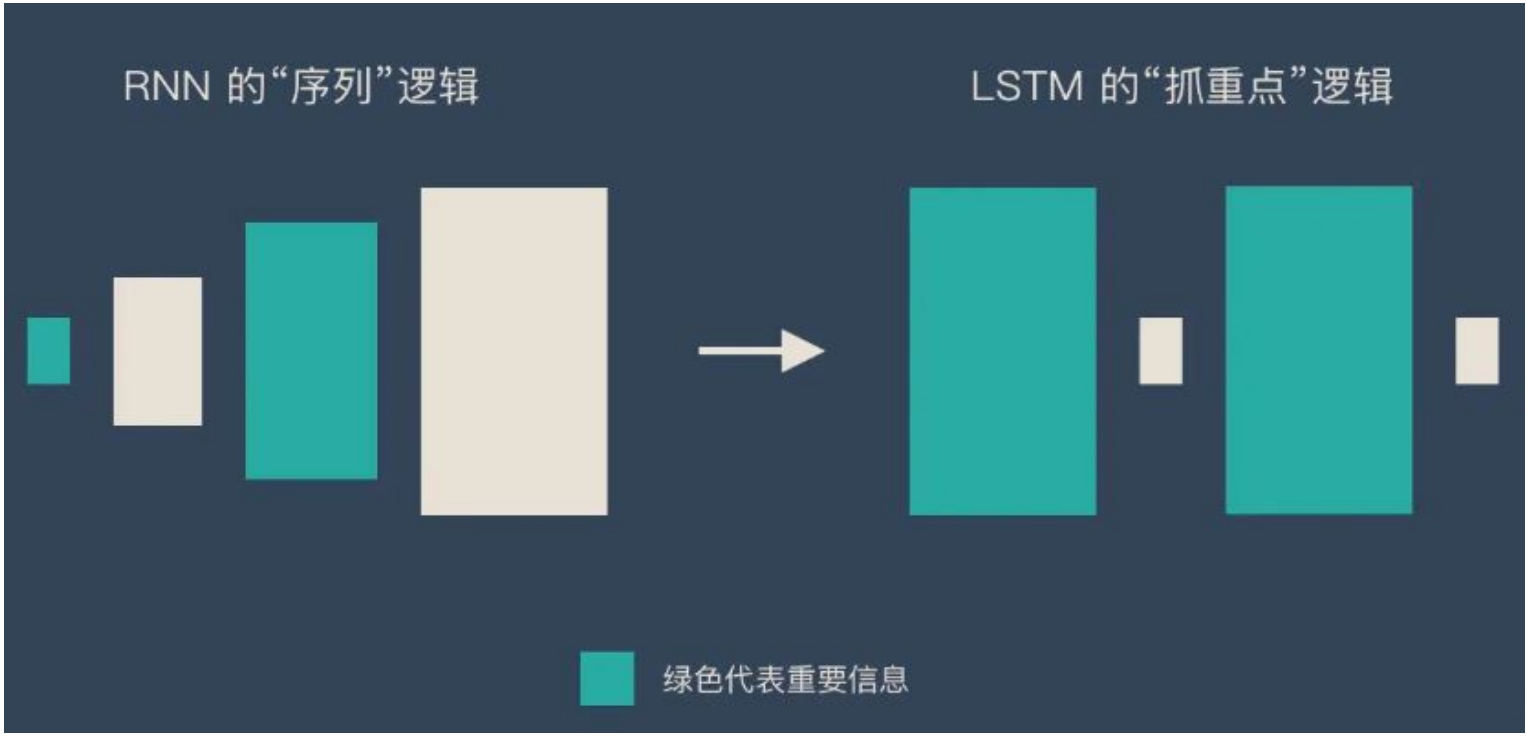
$$
\begin{aligned}
h_t &= f_1(h_{t-1}, x_t), \quad \forall t \in [T] \\
h_{T+t} &= f_2(h_{T+t-1}, \hat{y}_{t-1}), \quad \forall t \in [M] \\
\hat{y}_t &= g(h_{T+t}), \quad \forall t \in [M]
\end{aligned} \tag{4}
$$




Encoder RNN
I
am
a
person
_GO
Je
suis
une
personne
EOS
Decoder RNN

如何让 RNN 的记忆变长？

- RNN 是一种死板的逻辑，越晚的输入影响越大，越早的输入影响越小，且无法改变这个逻辑。
- Long Short Term Memory (LSTM) 做的最大的改变就是打破了这个死板的逻辑，而改用了一套灵活了逻辑——只保留重要的信息。

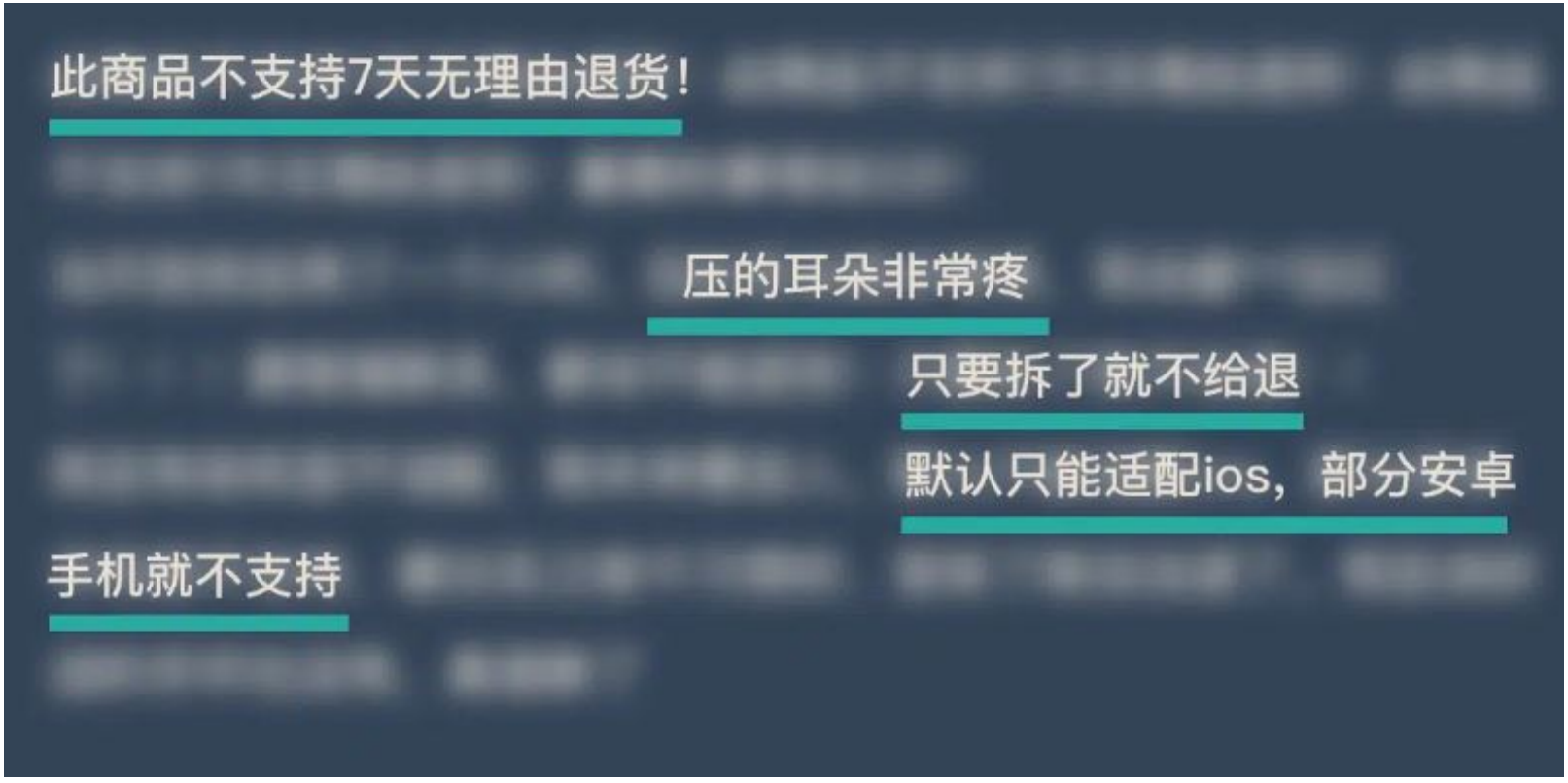


LSTM 的能力

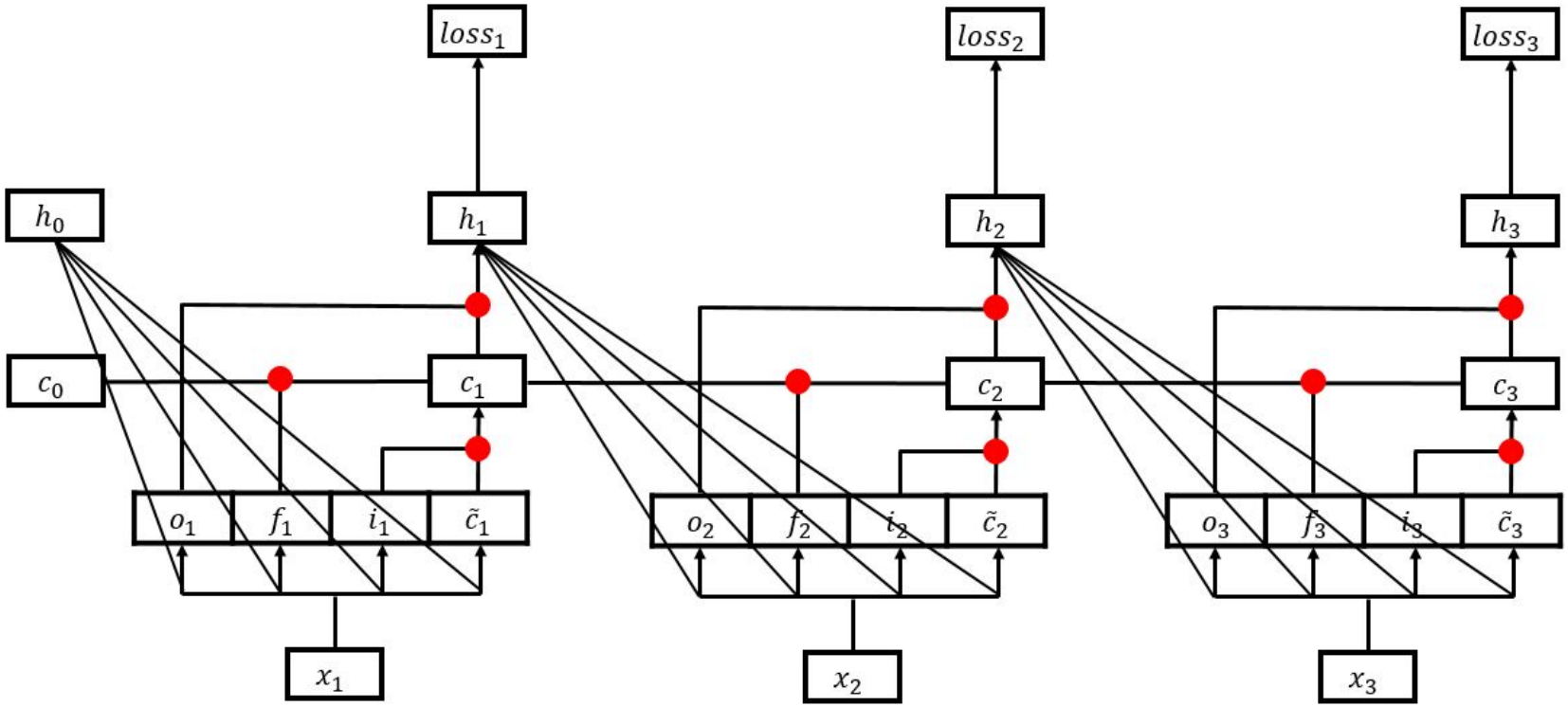
此商品不支持7天无理由退货！此商品不支持7天无理由退货！此商品不支持7天无理由退货！此商品不支持7天无理由退货！重要的事情说3次！

当天到货后用了一个小时，压的耳朵就非常疼，耳朵都**压红了！！！跟客服联系，都说不能退货！只要拆了就不给退！！

而且有些机型不适配，我本来要送人，默认只能适配ios，部分安卓手机就不支持，建议买之前千万想好，签收了就没法退了。而且说好送的手环也没有，真是醉了


此商品不支持7天无理由退货！
压的耳朵非常疼
只要拆了就不给退
默认只能适配ios，部分安卓
手机就不支持

为什么 LSTM 能成功呢？



模块四：Transformer 与大模型 (LLM) 的底层逻辑

核心技术解码（认知与语言革命）：

- 机制：“注意力机制”（Attention）——让 AI 在汪洋大海般的上下文中瞬间抓住重点。
- 本质：“大力出奇迹”，算力与数据达到阈值后，AI 涌现出了类似人类的“逻辑推理”与“生成能力”。

Transformer 解决什么语言问题?

Word Embedding

小明喜欢吃苹果　　小红喜欢吃橘子

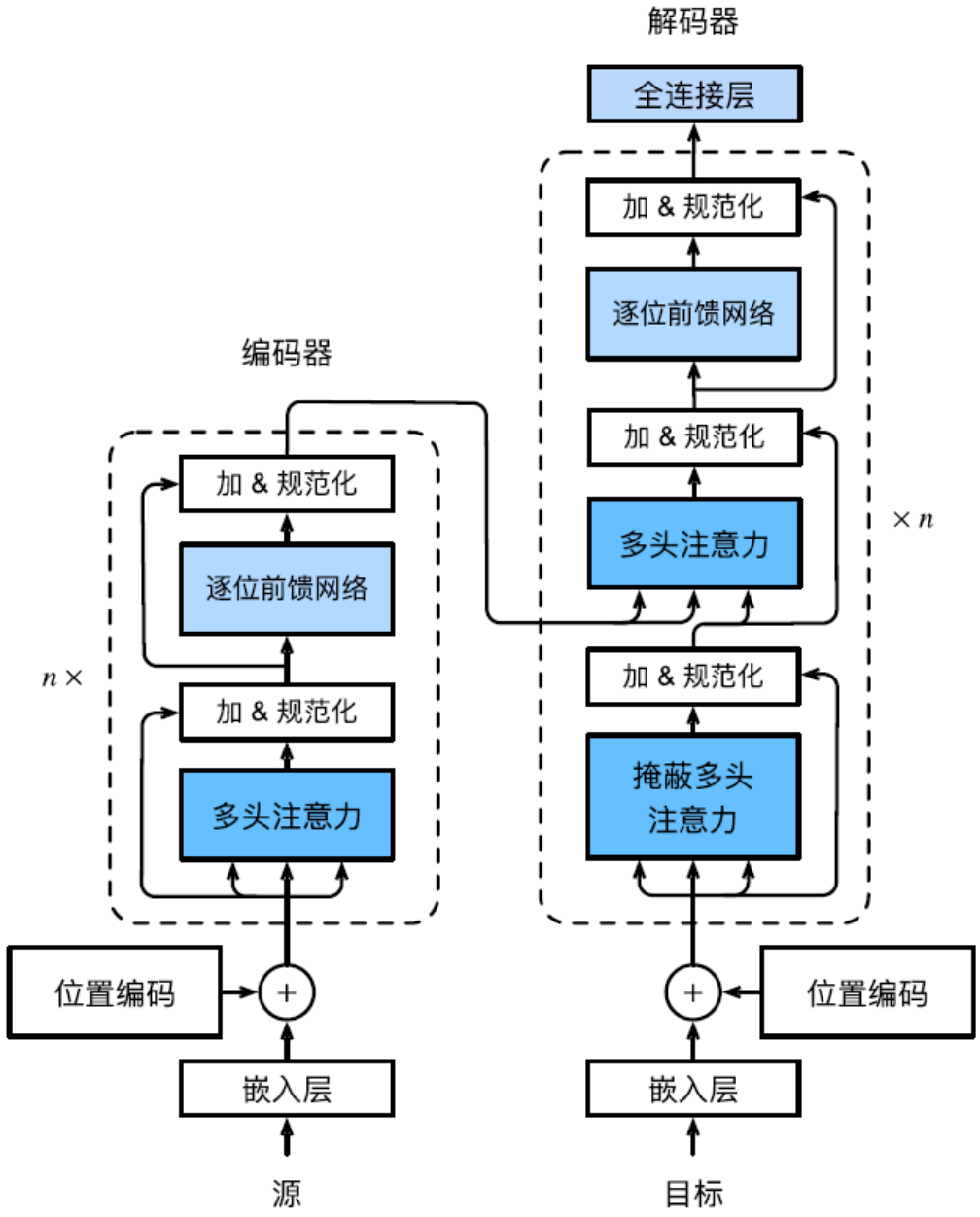
Word Position

曼联大胜曼城　　曼城大胜曼联

Word Association

春天到了，大自然生气勃勃　　小明考砸了，妈妈很生气

Transformer 模型架构



Transformer 基于如下模块的搭建。

0 嵌入层及位置编码
1 多头注意力 (masked)
2 逐位前馈网络
3 加 & 规范化层
a 编码器: (0 + (1 + 3 + 2 + 3)*n)
b 解码器: (0 + (1 + 3 + 1 + 3 + 2 + 3)*n)

- Transformer 的主要思想、贡献：

1. 注意力机制 –> 学习整个句子的内容。
2. 位置编码 –> 学习单词顺序。
3. 解码器和编码器 –> 让翻译变成现实。
4. 多头 –> 并行。

从 Transformer 到大模型

- Transformer 模型自 2017 年由 Vaswani 等人首次提出以来，已经成为自然语言处理（NLP）领域的核心技术之一。它的提出标志着深度学习在处理序列数据方面的一个重要突破，特别是在机器翻译、文本摘要、问答系统和文本生成等任务中取得了显著的成绩。以下是从 Transformer 到大模型的发展历程的简要概述.


分布式理论词袋模型
N-Gram模型
分布式表示
NNLM
RNNLM
Word2Vec
GloVe
CoVe
GPT、
EMLo、
BERT
XLNet、
RoBERTa、
GPT2、 T5、
ERNIE、
DistilBERT、
M-BERT、
VideoBERT
ELECTRA、
GPT3、
ALBART、
BART
Switch
Transform
er、Pangu、
Wudao、
M6
1948
1954
1986
2003
2010
2013
2014
2017
2018
2019
2020
2021
早期统计语言模型
基于神经网络的词嵌模型
基于深度学习的预训练模型

- Transformer 的诞生： 2017 年，"Attention Is All You Need" 论文中提出了 Transformer 模型，这是一种完全基于注意力机制的架构，摒弃了之前序列模型中常用的循环神经网络（RNN）结构。
- BERT 的出现： 2018 年，Google 的 BERT（Bidirectional Encoder Representations from Transformers）模型发布，它利用 Transformer 的编码器（Encoder）部分，通过大量文本的预训练，学会理解语言的深层含义。
- 多模态模型： 2021 年，CLIP（Contrastive Language–Image Pre-training）由 OpenAI 提出，这是一个多模态模型，能够理解图像和文本之间的关系。

GPT 系列发展历程（2018–2025.01）

- GPT 系列：
  1. 2018 年，OpenAI 发布 GPT（Generative Pre-trained Transformer），生成式预训练 Transformer 模型，用于文本生成任务。
  2. 2019 年，GPT-2 发布，模型规模更大，参数更多，生成文本能力更强。
  3. 2020 年，GPT-3 发布，拥有 1750 亿参数，可处理复杂文本任务，甚至完成简单编程。
  4. InstructGPT：在 GPT-3 基础上通过代码训练和人类偏好对齐，提升性能与安全性。
  5. ChatGPT：2022 年 11 月发布，基于 GPT 的对话服务，展现世界知识、复杂求解、多轮追踪、价值观对齐能力。
  6. GPT-4：2023 年 3 月发布，首度扩展至图文双模态，展现通用人工智能潜力。
  7. GPT-4o：2023 年 5 月发布，多模态大模型，支持文本、音频、图像的任意组合输入与输出。
  8. GPT-o1：2024 年 12 月发布，在科学、编程、数学等领域表现优异，具备长思维链与自适应计算能力。
  9. o3-mini：2025 年 1 月发布，成本高效的推理模型，支持低、中、高三档推理强度，在 STEM 任务中优于 o1-mini。

GPT 系列最新进展（2025.02–）

- 2025 年及后续重要更新：
  10. GPT-4.5：2025 年 2 月发布，OpenAI 最大规模模型，计算效率较 GPT-4 提升 10 倍，幻觉率降至 37.1%，定位为“非思维链”时代收官之作。
  11. GPT-4.1 系列：2025 年 4 月发布，支持百万 token 上下文，编程任务 SWE-bench 得分 54.6%，指令遵循显著提升。
  12. GPT-5：2025 年 8 月发布，统一架构整合推理模块与实时路由系统，免费开放，Pro 用户可享进阶版本。
  13. GPT-5.1：2025 年 11 月发布，GPT-5 的首次升级版本。
  14. GPT-5.2：2025 年 12 月发布，提供 Instant、Thinking、Pro 三种模型，强化智能体工具调用与长文本处理，被 OpenAI 称为“最聪明的通用模型”，幻觉率较 5.1 降低约 30%。
  15. GPT-5 后续规划：OpenAI 计划持续整合推理与非推理能力，推动模型自主判断何时使用深度思考、何时快速响应，向通用人工智能（AGI）持续演进。

通义千问 (Qwen) 发展史：Qwen 2 → Qwen 3.6

- **Qwen 2 & 2.5 系列 (2024 年中 - 2025 年初)**
  - Qwen 2 (24 年 6 月)：基础架构优化，全面开源并大幅增强多语言与长文本能力。
  - Qwen 2.5 (24 年 9 月起)：推出 72B 模型，提升代码与数学能力；百万上下文、多模态，具备深度推理能力等。
- **Qwen 3 系列 (2025 年 4 月)**
  - 架构与规模跨越：提供 Dense 与 MoE 版本 (最高达 235B 参数)，基于 36T Tokens 预训练，覆盖 119 种语言与方言。
  - 混合推理机制：创新性地引入“思考模式 (Thinking)”与“非思考模式”，兼顾逻辑深度与极速响应。
- **Qwen 3.5 系列 (2026 年 2 月)**
  - 极简显存与原生多模态：推出 397B 规模的开源旗舰 (仅激活 17B)，采用 Sparse MoE 与 Gated Delta Networks 混合架构。
  - 端侧无缝部署：同步发布高能效 Small 系列 (0.8B - 9B)，支持在手机等端侧设备上零成本离线运行。
- **Qwen 3.6 系列 (2026 年 4 月)**
  - 全能与开源双轨：推出商用闭源的全模态 **3.5-Omni** 与最强旗舰 **Qwen 3.6-Plus**。
  - 消费级大杀器：**Qwen 3.6-35B-A3B** 模型，主打在 32GB 显存消费级显卡上实现复杂的智能体工作流。

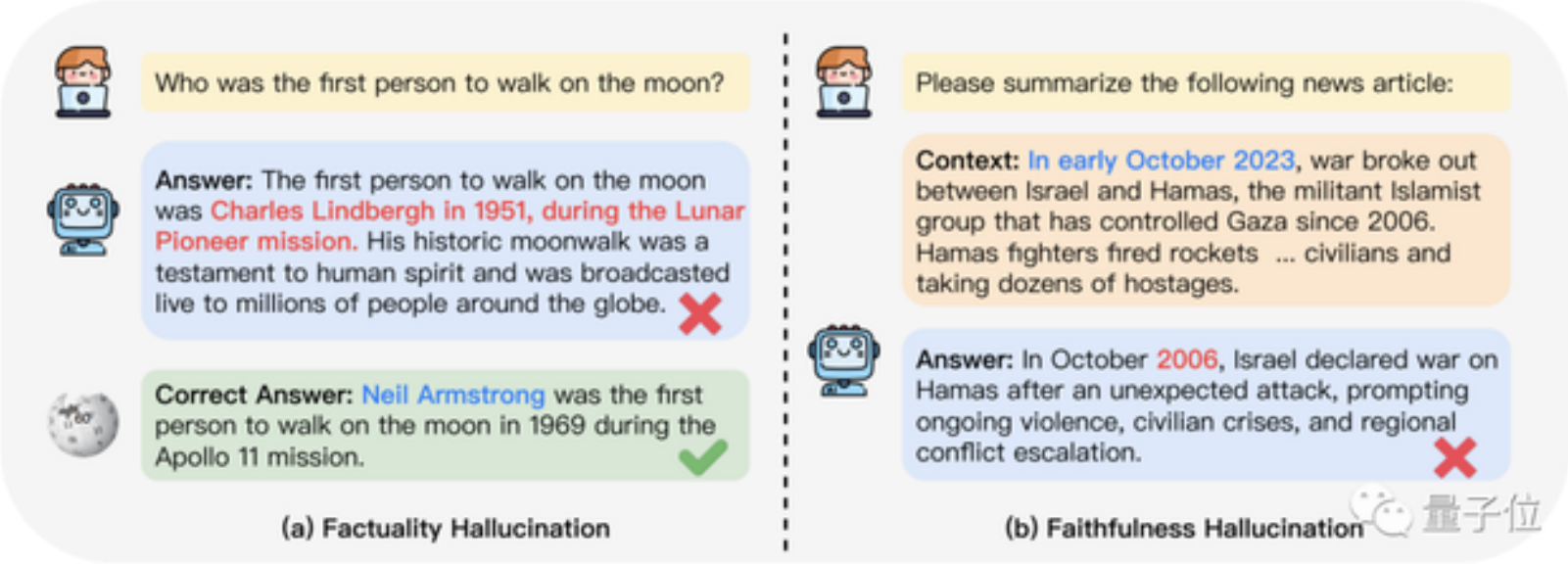
LLM meets Uncertainty

- 涌现：在小模型上没有出现，但是在大模型上出现的不可预测的能力，就被称为 Emergent Capabilities.
- 幻觉：AI 模型会产生完全捏造的信息，既不准确也不真实 (Hallucination).
- 生成的答案是随机的!

如何理解“涌现”能力？

- **定义：** 涌现能力指的是在大模型中，当模型的规模（包括计算量、模型参数或数据集大小）达到一定程度时，某些能力会突然出现，性能显著提升。这种能力在小模型中并不存在，而是在大模型中表现出来的.
- **非线性增长与临界点：** LLM 的涌现能力并非随着模型规模的线性增长而平滑提升，而是在达到某个临界点后急剧增强.
- **不可预测性：** 尽管我们可以通过实验观察到涌现能力，但目前仍难以准确预测一个模型将在何时、以何种形式展现出新的能力.
- **强大的泛化能力：** 涌现出的能力不仅能很好地处理训练数据中的任务，而且能泛化到新的、未见过的任务上.

AI 幻觉


Who was the first person to walk on the moon?
Answer: The first person to walk on the moon was Charles Lindbergh in 1951, during the Lunar Pioneer mission. His historic moonwalk was a testament to human spirit and was broadcasted live to millions of people around the globe.
Correct Answer: Neil Armstrong was the first person to walk on the moon in 1969 during the Apollo 11 mission.
(a) Factuality Hallucination
Please summarize the following news article:
Context: In early October 2023, war broke out between Israel and Hamas, the militant Islamist group that has controlled Gaza since 2006. Hamas fighters fired rockets ... civilians and taking dozens of hostages.
Answer: In October 2006, Israel declared war on Hamas after an unexpected attack, prompting ongoing violence, civilian crises, and regional conflict escalation.
(b) Faithfulness Hallucination
量子位

如何理解“幻觉”问题？

- **大语言模型幻觉**（Large Language Model Hallucination）是指大型语言模型（如 GPT 系列、BERT 等）在生成文本或执行任务时可能会出现的错误或不准确的输出。这种现象类似于人类的认知幻觉，即模型生成的内容可能与现实世界的事实不符，或者在逻辑上存在矛盾。

可能发生的原因

1. 训练数据的局限性
2. 泛化能力不足
3. 上下文理解不足
4. 知识更新滞后
5. 模型自信过高
6. 缺乏常识推理
7. 对抗性和误导性输入

事实性幻觉

- 事实性幻觉 (Factuality Hallucination)，是指模型生成的内容与可验证的现实世界事实不一致。
- 比如问模型“第一个在月球上行走的人是谁？”，模型回复“Charles Lindbergh 在 1951 年月球先驱任务中第一个登上月球”。实际上，第一个登上月球的人是 Neil Armstrong。
- 事实性幻觉又可以分为事实不一致（与现实世界信息相矛盾）和事实捏造（压根没有，无法根据现实信息验证）。

忠实性幻觉

- 忠实性幻觉 (Faithfulness Hallucination)，则是指模型生成的内容与用户的指令或上下文不一致。
- 比如让模型总结今年 10 月的新闻，结果模型却在说 2006 年 10 月的事。
- 忠实性幻觉也可以细分，分为指令不一致（输出偏离用户指令）、上下文不一致（输出与上下文信息不符）、逻辑不一致三类（推理步骤以及与最终答案之间的不一致）。

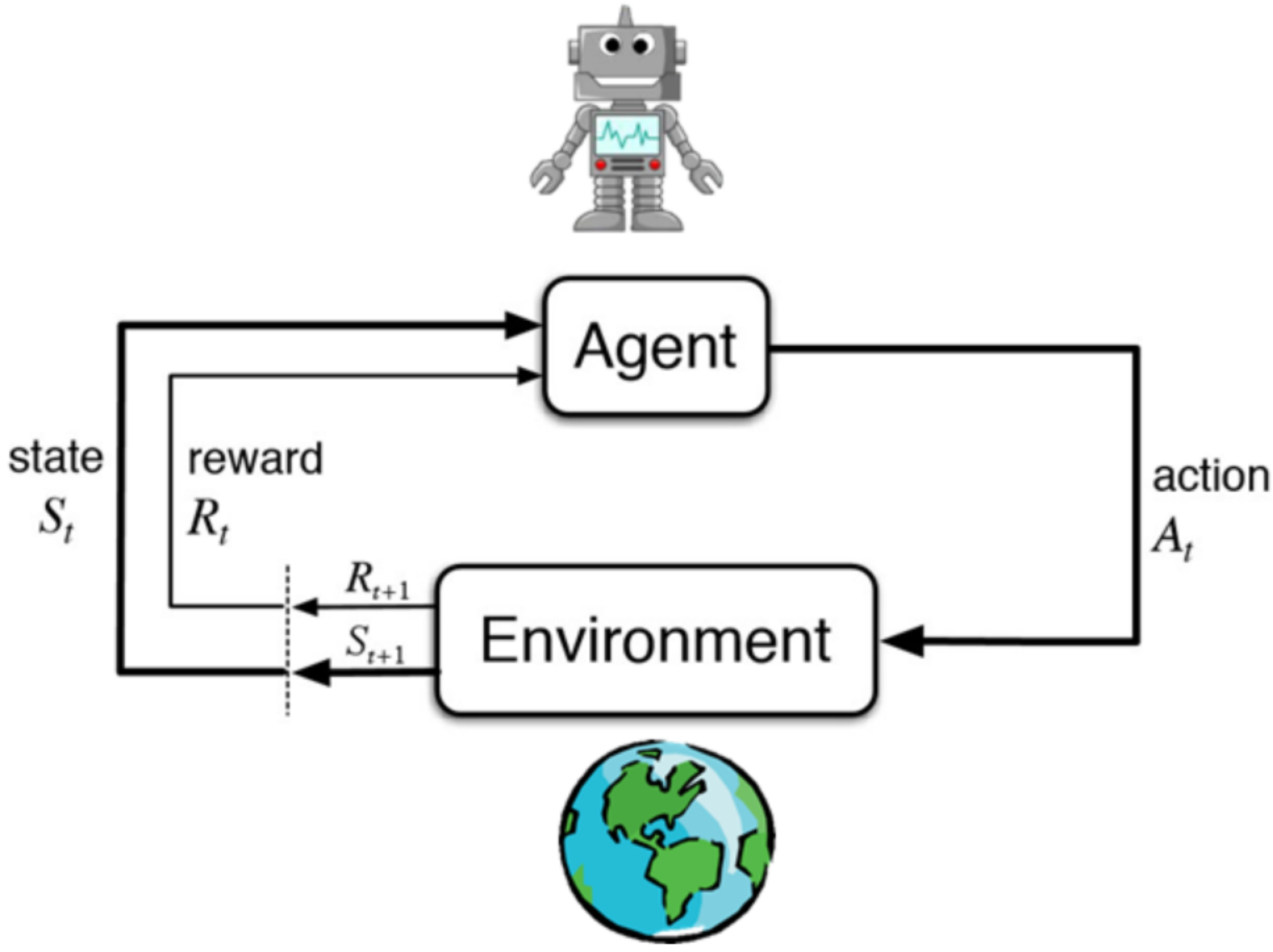
如何利用统计方法消除幻觉？

- 我们可以使用任意的大模型对文本进行修改，来得到更“真实”的文本。
- Deepseek，腾讯元宝，豆包，KIMI 都可以实现。

模块五：强化学习 (RL) 的底层逻辑

核心技术解码（在复杂世界中“打怪升级”）：

- 机制：环境互动 $\rightarrow$ 采取行动 $\rightarrow$ 获得奖惩反馈（Trial and Error）。
- 本质：AI 不需要人类告诉它标准答案，它为了实现“全局收益最大化”，能够自己摸索出最优策略（如 AlphaGo）。


Agent
state
S_t
reward
R_t
action
A_t
R_{t+1}
S_{t+1}
Environment

强化学习的目标是什么？

- 强化学习的目标就是寻找一个策略，使得回报的期望最大化。这个策略称为最优策略 (optimum policy)。
- 一个好的策略是最大化回报，而不是最大化当前的奖励。
- 例如：下棋的时候，你的目标是赢得一局比赛（回报），而非吃掉对方当前的一个棋子（奖励）。
- 核心：exploration (探索) vs exploitation (剥削).

为什么我们要关注一道数学题？

- **不是为了做题，而是为了测试“逻辑推理能力”**
- 过去的大模型（如早期 ChatGPT）：像是一个“熟读唐诗三百首”的文科生。依靠概率接龙，擅长总结和写文案，但面对需要严密逻辑的场景常常“一本正经地胡说八道”。
- **AIME（美国数学邀请赛）的象征意义**：
  - 代表了极度复杂的**多步逻辑推导**和**极低的容错率**（错一步，全盘皆输）。
  - 如果 AI 能解决它，意味着 AI 开始具备解决真实世界复杂商业问题（如供应链规划、财务推演、战略分析）的潜力。

核心观点

AI 正在从“凭借直觉的快思考”走向“深思熟虑的慢思考”。

来看一道测试题（模拟 AIME 逻辑的代数题）

题目：求所有使得 $\frac{n^2+3n+5}{n+1}$ 结果为整数的整数 n 的和。

这道题的考点：
不能靠“死记硬背”或“随便猜几个数”，必须把分子拆解（多项式除法），转化为严密的逻辑推导。我们来看看 AI 在经历强化学习（RL）前后 的表现差异。

强化学习前（RL 前）：靠直觉的“快思考”

未经过 RL 训练的 AI（类似传统大模型）的回答：

“为了让结果是整数，我们可以尝试代入几个数字猜一下。
假设 $n = 1$，结果是 $(1 + 3 + 5)/2 = 4.5$，不是整数。
假设 $n = 2$，结果是 $(4 + 6 + 5)/3 = 5$，**这是一个整数！**
假设 $n = 3$，结果是 $(9 + 9 + 5)/4 = 23/4 = 5.75$，不是。
假设 $n = 4$，结果是 $(16 + 12 + 5)/5 = 33/5 = 6.6$，不是。
看来只有 $n = 2$ 符合条件。
最终答案：这些整数的和是 2。” **（错误！）**

商业映射：这就好比一个没有经验但很自信的员工，遇到问题“拍脑袋”抽样，浅尝辄止，没有结构化思维，最终给老板提供了一个漏掉无数隐患（负数情况）的错误决策。

强化学习后（RL 后）：深思熟虑的“慢思考”（Qwen 推理模型）

经过 RL 训练的 Qwen 的内心独白（Test-Time Compute）：

[思考过程开始]

1. 题目要求 $\frac{n^2+3n+5}{n+1}$ 为整数。直接代入数字容易漏算，我需要进行代数变形。
2. 尝试将分子写成含有 $(n+1)$ 的形式：
$n^2+3n+5=(n^2+2n+1)+n+4=(n+1)^2+(n+1)+3$。
3. 所以原式 $=\frac{(n+1)^2+(n+1)+3}{n+1}=(n+1)+1+\frac{3}{n+1}=n+2+\frac{3}{n+1}$。
4. 因为 n 是整数，所以 $n+2$ 必然是整数。要让整体是整数，**$\frac{3}{n+1}$ 必须是整数 **！
5. 这意味着 $(n+1)$ 必须是 3 的约数。3 的约数有四个：$1,-1,3,-3$。
6. 分别求解 n：
   若 $n+1=1$，则 $n=0$
   若 $n+1=-1$，则 $n=-2$
   若 $n+1=3$，则 $n=2$
   若 $n+1=-3$，则 $n=-4$
7. 题目求所有可能的 n 的和：$0+(-2)+2+(-4)=-4$。

[思考过程结束]

最终答案：-4。 （完美正确！）

对比的启示：强化学习（RL）赋予了 AI 什么？

从刚才的例子可以看出，Qwen 学会的不仅仅是数学公式，而是**一套解决复杂问题的方法论**：

- **学会了拆解**：把复杂问题（二次多项式）转化为简单问题（找约束）。
- **穷举与严密性**：不再盲人摸象，而是通过逻辑锁定了所有可能性（包括人类容易忽略的负数）。
- **自我反思与验证**：在后台的长思维链（Chain of Thought）中，AI 自己推导、自己验证，直到逻辑闭环才给出结论。

它是怎么练成这种能力的？——类比企业的人才培养

Qwen（通义千问）攻克数学难题的过程，非常像企业培养一位顶尖的“战略分析师”：

1. **第一阶段：海量阅读（预训练 Pre-training）**
   ——招募高智商、博览群书的应届生。建立了语言和公式基础。
2. **第二阶段：名师带徒（监督微调 SFT）**
   ——给 *AI* 喂养大量人类专家的标准解题过程，学习公司的 *SOP*。
3. **第三阶段：商战演练（强化学习 RL）[最关键的蜕变]**
   ——不再教它怎么做，而是设定规则（对了重奖，错了惩罚）。让 *AI* 自己在千万次试错中，自己“悟”出了多项式除法、回溯和自我检查的诀窍！

杀手锏：测试时计算（Test-Time Compute）

定律转换：从“力大砖飞”到“深思熟虑”

过去我们认为：大模型参数越大（脑容量越大）才越聪明。
现在 Qwen 证明了：**给 AI 的思考时间越长，它的决策质量越高**。

- 在解决真正的 AIME 竞赛题时，Qwen 会在后台生成长达数万字的“内心独白”。
- 它可以尝试 3 种不同的解法，发现第一种走不通时自动“退回原点”，用另一种方法互相印证。
- **启示**：在企业应用中，针对高价值决策（如投资并购、供应链排产），我们可以允许 AI 在后台推演几个小时甚至一天，以换取人类无法企及的严密决策方案。

对企业管理者的商业启示

面对具备了“AIME 级别推理能力”的 AI，我们应该怎么做？

- **重新定义 AI 的角色**：
  不要再把 AI 当成只会写漂亮场面话的“文秘”。它正在进化为能处理多变量、低容错率任务的“理科生参谋长”。
- **数据资产的新价值（思维链数据）**：
  企业内部高管做决定的“会议纪要、推导逻辑、失败复盘报告”比单纯的“结果财务报表”更有价值。这是用 RL 训练企业专属 AI 专家的黄金燃料。
- **组织架构的变革**：
  未来的企业决策可能变为：“人类提出战略目标 $\rightarrow$ AI Agent 生成长逻辑链的多重推演方案 $\rightarrow$ 人类拍板”。

未来的数字员工——AI Agent (智能体)

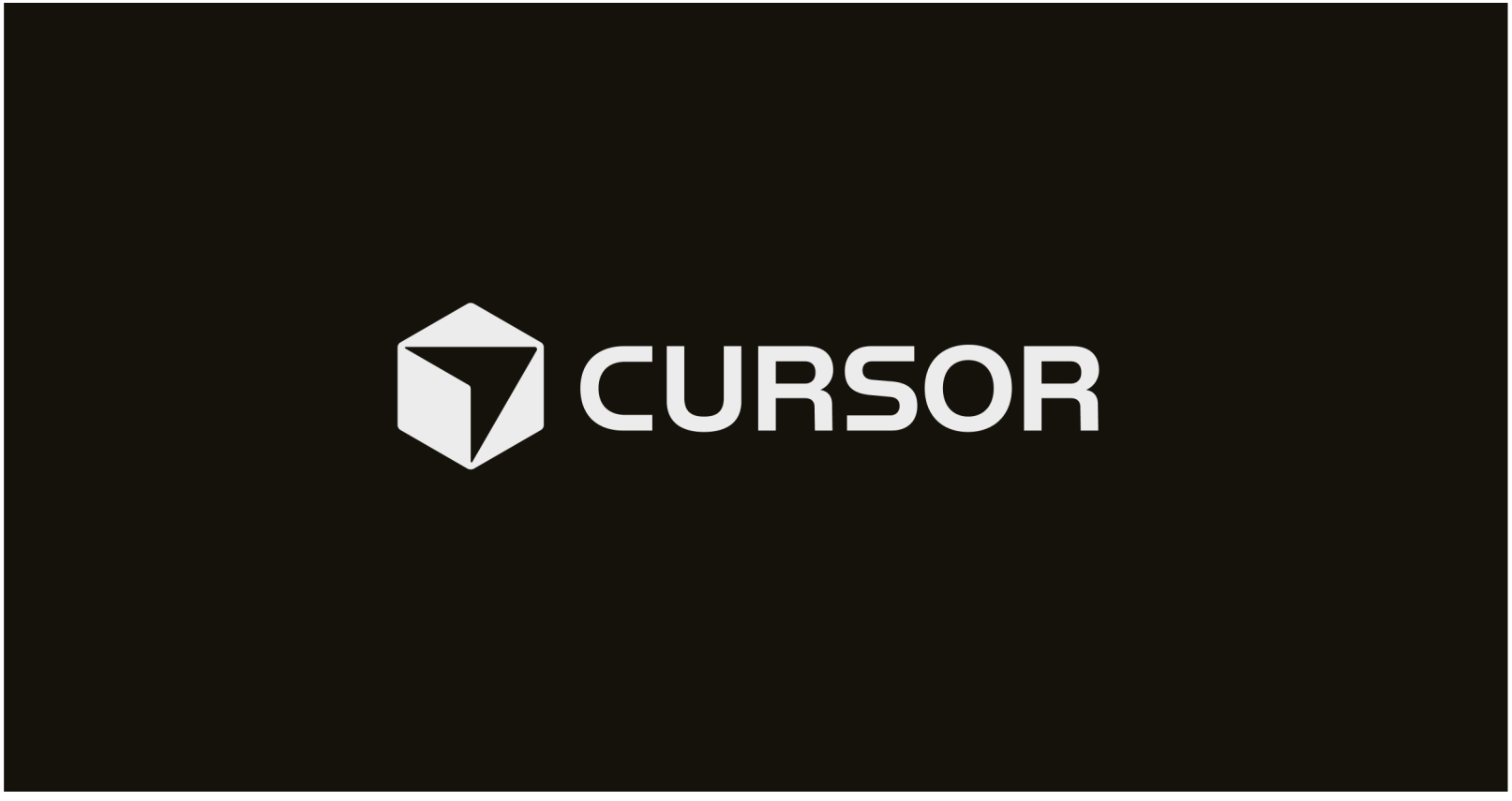
技术拼图的终局：Agent ＝大模型 (大脑) ＋感知/记忆＋规划 (思维链) ＋使用工具 (手脚)

- **大语言模型 (LLM) 只是坐在椅子上的大脑，而 Agent 是拥有手脚、能自主执行任务的实体。**
- 演进路线：从 Copilot（需要人一步步发指令）→ Auto-Agent（只需人给一个大目标，AI 自己拆解并执行）。

战略拷问

如果您可以雇佣 1000 个 24 小时工作、不要社保、自带全球顶尖知识库的“数字大学生”，您的公司形态会发生什么变化？

Dessert


CURSOR

Take Away

- 新技术 – 看计算机会议
- 高观点 – 四大核心版块
- 主要知识 – MLP, CNN, RNN, Transformer
- 好伙伴 – 大模型与 Agent

谢谢，提问环节？